Universidad de Castilla-La Mancha

Escuela Superior de Ingeniería Informática

Departamento de Sistemas Informáticos
Programa Oficial de Postgrado en Tecnologías Informáticas Avanzadas

Trabajo Fin de Máster

Bases de Datos de Series Temporales: Representación y Consultas

Septiembre de 2012

Alumno: Antonio Moreno García

Tutor: Dr. D. Juan Moreno García

Índice general

1

Introducción

Las series de tiempo son un motivo de estudio desde hace décadas ya que están presentes en muchos de los tipos de datos almacenados en diferentes bases de datos, por ejemplo, bases de datos financieras, médicas o científicas. La mayoría de estas bases de datos son muy extensas, y por lo tanto difíciles de tratar. Es muy importante encontrar una forma de representarlas que permita realizar cualquier tipo de operación de forma rápida y que permita almacenarlas con un ahorro de memoria. Por ejemplo, una base de datos de un sensor que realice una toma de un valor real cada 5 minutos generaría 288 valores al día, lo que implica 105120 valores al año, esto sólo para un sensor.

La representación debe ser robusta, con un error mínimo, para representar correctamente el conocimiento de la serie y para trabajar con ella de una forma simple y rápida. Además debe permitir trabajar en problemas que habitualmente manejan series temporales de datos [8], como por ejemplo: matching [66], previsión [25, 4, 67, 93, 58, 97, 91, 68], consultas [88], clustering [28, 48, 29], data mining [21, 8], etc.

Una solución frecuente en muchas investigaciones consiste en dividir la secuencia en subsecuencias que se representan mediante funciones matemáticas que permiten un proceso de discretización de la serie. Una representación discreta permite trabajar con series tomadas de bases de datos de una manera más sencilla, ya que el tamaño de los datos almacenados es bastante inferior en comparación con los valores reales tomados.

Por otro lado, este tipo de representaciones genera gran cantidad de incertidumbre, dado que, aparte de la propia incertidumbre provocada por la captura de los datos, se comete un error al representar un conjunto de datos temporales mediante una función matemática que todavía aporta más incertidumbre. Por ello, una buena solución para este tipo de representación puede ser la utilización de la lógica difusa dentro de este ámbito. Ya existen algunos trabajos en esta línea como se mostrará en la Capítulo 3.

Un problema muy importante sobre las series de tiempo es el de búsqueda de información o consultas sobre las mismas. Su importancia radica en:

- La implantación tecnológica de las BDs en la sociedad actual: Las bases de datos están ampliamente implantadas en todos los sectores de la sociedad y se sirven de entornos informatizados que manejan gran cantidad de información. Además, ésta es una tendencia que va a más.
- La sed de conocimiento de la sociedad actual: En esta sociedad informatizada las personas desean conocer cosas nuevas y rescatar información de diferentes fuentes (BDs generalmente) de una forma rápida, y con resultados claros y concisos, tanto a nivel profesional como a nivel de entretenimiento. Esta información se obtiene mediante consultas en BDs usualmente por Internet y a través de aplicaciones específicas y/o buscadores.

1.1. Objetivos.

El primer objetivo que se pretende alcanzar con la elaboración del presente Trabajo de Fin de Máster consiste en un estudio del estado del arte acerca de las series de tiempo. Se estudiarán las diferentes formas de representación de las mismas en la bibliografía, destacando las representaciones más aceptadas, robustas y eficientes. También se estudiarán los diferentes métodos de consultas sobre estas representaciones.

El segundo objetivo del Trabajo de Fin de Máster estriba en el planteamiento de dos representaciones eficientes de las series de tiempo que permitan la representación de las series de una forma más eficiente y robusta para la realización de consultas robustas y eficientes en BDs temporales. Se abordarán dos nuevos tipos de representación de las series de tiempo:

1. Representación de la serie mediante un conjunto ordenado de segmentos que serán obtenidos mediante un método totalmente automático. Se deberá tratar la forma de representación y el método para obtenerla.
2. Representación de la serie utilizando lógica difusa en la representación de los segmentos, lo que gestionará la incertidumbre anteriormente comentada. El proceso consiste en la creación de una novedosa estructura de representación de cada uno de los segmentos obtenidos en el proceso anterior. En resumen, se transforma el conjunto ordenado de segmentos en un conjunto ordenado de estructuras de más alto nivel. Este tipo de representación facilitará el posterior procesamiento en muchos tipos de problemas.

El tercer objetivo será trabajar en la búsqueda de información en estas representaciones. Se realizará una primera aproximación a consultas sobre la serie representada como segmentos. En funcion de estos objetivos se plantean los siguientes objetivos parciales:

1. Estudio de los modelos de representación de series de tiempo.
2. Estudio de los métodos de búsqueda de información en estas series.
3. Obtención de un método de conversión de una serie temporal a un conjunto ordenado de segmentos. Debe ser automático y lo menos parametrizado posible.
4. Diseño de una estructura difusa para la representación de un segmento.

5. Método de conversión del conjunto ordenado de segmentos a un conjunto ordenado de la estructuras difusas.

6. Realización de consultas sobre el conjunto ordenado de segmentos.

Así, el alcance de este trabajo de investigación es el de realizar un estudio de los modelos de representación eficiente de series de tiempo, así como de busqueda de información en series temporales. Además se consigue un novedoso método para la representación de series temporales mediante conjuntos ordenados de segmentos. También se obtiene una representación difusa del modelo anterior. Y finalmente se realizan consultas sobre la primera representación.

1.2. Estructura del Documento.

A continuación se presentan los contenidos y estructura de esta memoria.

En el capítulo 2 se exponen las asignaturas realizadas.

En el capítulo 3 se estudia el estado del arte en torno al cual gira la línea de investigación que se presenta en este documento. Los dos puntos principales de este capítulo son la representación e indexación de datos, y los métodos existentes para la búsqueda de patrones.

En el capítulo 4 se detalla el tema central de la investigación. Se presentarán las dos representaciones propuestas y el método de consulta sobre una de las representaciones.

En el capítulo 5 se presentan otros resultados preliminares relacionados con la línea de investigación propuesta.

En el capítulo 6 se exponen las conclusiones obtenidas como resultado del trabajo realizado y los trabajos futuros.

Finalmente, se presenta la bibliografía.

En el Anexo A, se detalla un breve Curriculum Vitae del alumno.

2

Asignaturas cursadas en el Máster

En este apartado se encuentran los resúmenes de las asignaturas del Master realizadas durante el año 2009 (2º cuatrimestre del curso académico 2008/09), principios del 2010 (2º cuatrimestre del curso académico 2010/11) pertenecientes al Máster en Tecnologías Informáticas Avanzadas de la UCLM. Se incluye además una breve descripción de las asignaturas del ya extinguido programa de Doctorado que me fueron convalidadas.

2.1. Tecnología Software Orientada a Objetos.

Impartido por los Doctores María Dolores Lozano Pérez, Elena María Navarro Martínez y Victor Ruiz Penichet.

2.1.1. Descripción.

Se centra principalmente en el modelado de sistemas para un desarrollo rápido y coherente de aplicaciones. Se eleva el nivel de abstracción en cuanto a la forma de desarrollar software, consiguiendo cada vez más cercanía con el lenguaje natural. Exponiéndose también las nuevas tecnologías de modelado y desarrollo orientado a objetos, centrado principalmente en MDA (Model Driven Architecture) y MDE (Model Driven Enviroments). Se introducen los conceptos relacionados con los Lenguajes Específicos de Dominio y se contrastan con los Lenguajes de propósito general.

Se estudiaron modelos y plataformas relacionados con la generación semi-automática y automática de código fuente. También se detallaron las posibles traducciones entre diferentes modelos y diferentes tipos de compiladores de modelos. Se explicaron diferentes metodologías de modelado como UML, CWM, MOF, etc. Además en este tipo de tecnologías los estándares tienen una función fundamental, dando consistencia a los diferentes modelos.

2.1.2. Trabajo realizado.

Se hizo un estudio para evaluar los trabajos realizados con la gestión de datos temporales, tanto a nivel de modelos de datos como de interfaz de usuario.

2.1.3. Conclusiones del alumno.

El modelado de sistemas facilita y da consistencia a todo el proceso de desarrollo de aplicaciones, facilitando también el periodo de mantenimiento de éstas. Se cuenta además con la gran ventaja de la generación automática y semi-automática de código, con lo que al cambiar el modelo bastaría con rehacer el mismo, en lugar de rehacerlo y posteriormente hacer lo mismo con el código.

La desventaja de realizar aplicaciones mediante modelado es que hay que realizar una importante labor en la creación de módelos para la aplicación que se desea crear. Cuando se trata de una aplicación puntual que no se va a difundir en varias plataformas (PDAs, Mobiles , . . .), éste esfuerzo no queda debidamente compensado en aplicaciones que no tengan un largo tiempo de vida.

2.2. Programación Internet con lenguajes declarativos multiparadigma.

Impartido por los Doctores Ginés Moreno, Pascual Julián Iranzo y Francisco Pascual Romero.

2.2.1. Descripción.

Tiene como objetivo presentar los fundamentos de los lenguajes declarativos multiparadigma, que integran la principales características de los lenguajes lógicos y funcionales puros, así como sus extensiones basadas en lógica difusa (fuzzy).

Durante el curso se expusieron distintos lenguajes de programación: PROLOG y FLOPER, y algunas aplicaciones, como por ejemplo, el uso en búsquedas en internet.

2.2.2. Trabajo realizado.

Como trabajo se implementó un prototipo en PROLOG para la ejecución de comparaciones de series temporales. Este prototipo toma como entrada una serie de datos, una consulta y el tipo de comparación (3 criterios: binario, Sin umbral y con umbral) a realizar, creando una lista resultado de ternas que contienen: posición dentro de la lista, lista utilizada en la comparación y valor numérico resultado de esta comparación. Este prototipo también se planteó a modo teórico en la extensión de PROLOG creada por la UCLM llamada FLOPER que permite el uso de lógica difusa.

2.2.3. Conclusiones del alumno.

Los lenguajes declarativos tienen la ventaja de que con pocas reglas se pueden crear sistemas de búsqueda relativamente complejos. Además, la desventaja que tradicionalmente tenían (la velocidad de ejecución) se está solucionando a pasos agigantados.

2.3. Asignaturas convalidadas.

Para completar el número de créditos necesarios para la obtención del Master convalidé las asignaturas del programa de Doctorado Técnicas Informáticas Avanzadas (Universidad Politécnica de Valencia) que había cursado en los años 1998 y 1999. A continuación se expone brevemente una descripción de las asignaturas superadas de acuerdo con el programa de Doctorado ya extinguido.

2.3.1. Computadores neuronales.

Este curso fue impartido por el Dr. Miguel Fernandez Graciani. Se presentaron las diferentes técnicas planteadas dentro del campo de las redes neuronales, así como sus posibilidades de aplicación. Para ello, tras el estudio de los principios de funcionamiento de los sistemas neuronales biológicos, se establecía una comparativa con los sistemas artificiales como forma de asimilación de los conceptos básicos en los que está basada esta tecnología.

Se introdujeron los módelos neuronales artificiales más básicos, como son el perceptron (entendido como unidad básica en una red reuronal), las propuestas de ADALINE y MADALINE; y las redes de retroalimentación (backpropagation), como respuesta surgida a los problemas de convergencia mostrados por los anteriores modelos. Se presentaron diferentes ejemplos donde la aplicación de estas técnicas puede ser útil, como son los problemas de reconocimiento del habla o en la clasificación de imagenes.

También se introdujeron diferentes técnicas de aprendizaje no supervisado, como los mapas auto-organizativos, empleados especialmente para la detección de grupos similares (clusters) sobre los datos de entrada. Estos presentan como ventaja su entrenamiento no supervisado, y tienen inconvenientes relativos al posterior procesamiento de los resultados para su interpretación en el contexto sobre el que se aplica.

2.3.2. Introducción a las algebras de procesos.

Este curso fue impartido por el Dr. Fernando Cuartero Gómez y tenía como objetivo principal estudiar los módelos algebraicos formales para el análisis de sistemas concurrentes. Para ello, plantea en primer lugar un estado del arte en cuanto a los modelos formales empleados en el análisis de sistemas concurrentes, para centrarse en el estudio de los modelos algebraicos.

A continuación se pasa a presentar, de forma general, conceptos fundamentales en relación con la sintaxis y semántica de los lenguajes formales, así como con los tipos de semántica (denotacional, operacional y ecuacional o algebraica).

Posteriormente se concreta el estudio de las álgebras de procesos en el modelo CSP

(lenguaje algebraico utilizado para describir sistemas compuestos que evolucionan de forma concurrente). De este modelo se exponen sus principales características: sintaxis, operadores, semántica denotacional y semántica operacional.

Por último, se introduce el algebra de procesos CCS, que presenta ciertas diferencias con CSP. Este modelo se estudia de forma mucho más breve, esbozando simplemente sus líneas maestras y contrastando éstas con las de CSP.

2.3.3. Técnicas de análisis de redes de Petri.

Este curso fue impartido por el Dr. Valentin Valero Ruiz y tenía como objetivo el conocimiento de las redes de Petri como herramienta de evaluación formal de sistemas concurrentes.

Comienza introduciendo los conceptos básicos relacionados con las redes de Petri marcadas. Posteriormente se estudian las propiedades de estas redes, las cuales se pueden clasificar en dos grandes grupos como son la seguridad y la vivacidad. Junto con estas propiedades se presentan diversas técnicas de verificación de las mismas, así como, el uso de cerrojos y trampas en el análisis de propiedades.

Además, se plantea la problemática del análisis de propiedades indecidibles, así como diversas técnicas de reducción de redes que nos permitirán simplificar las redes obtenidas para un determinado sistema haciéndolas de este modo más manejables.

Para finalizar el curso se introducen diversas extensiones del modelo de redes de Petri, entre los que podemos destacar las redes temporizadas y las redes coloreadas.

2.3.4. Arquitecturas paralelas.

El curso, impartido por el Dr. Francisco José Quiles Flor, tenía como objetivo presentar distintas arquiteturas paralelas propuestas como solución a las necesidades computacionales de la visión por ordenador.

El curso comienza describiendo los requisitos computacionales de la visión por computador y la necesidad de las arquitecturas paralelas dadas las limitaciones de la arquitectura Von Neumann. A continuación, se establece una clasificación de arquitecturas paralelas basada en los conceptos de corriente de instrucciones y corriente de datos, que da lugar a la división de la organización de computadores en: SISD (Single Instruction Single Data), MISD (Multiple Instruction Single Data) y MIMD (Multiple Instruction Multiple Data). El resto del curso se desarrolla mostrando las características y ejemplos de los distintos tipos de arquitecturas.

El curso incide en el concepto de paralelismo externo para analizar las distintas posibilidades que presentan las redes de interconexión de computadores en cuanto a topología (toros, mallas, hipercubos, redes Clos, redes de Benes, etc.) y encaminamiento. Se profundiza también en los conceptos de arquitectura y máquina vectorial, máquinas de memoria compartida y distribuida, máquinas de paso de mensajes y arquitecturas híbridas, analizando para cada una de estas filosofías sus ventajas, inconvenientes y limitaciones.

Además, se estudian detalladamente ejemplos de arquitecturas de computadores SIMD reales, como es el caso del CM-2 y el MasPar MP-1, y de propuestas como el proyecto de computador reconfigurable REMAP3 (Reconfigurable, Embedded Massively Parallel Proce-

sor Project), incluyendo en este último caso una introducción al lenguaje de microprogramación (microcode assembrer) y al lenguaje de alto nivel SMAL.

2.3.5. Redes de alta velocidad.

Este curso fue impartido por el Dr. Antonio José Garrido del Solo, tuvo como objetivo principal de introducir la técnologia ATM, así como su aplicacion en sistemas distribuidos.

Se estudiaron los diferentes aspectos relacionados con las redes de banda ancha y, en concreto, con la tecnología que los sustenta como es ATM. Principalmente tras la introducción de la problemática derivada de las redes de banda ancha en cuanto a su escalabilidad, así como las aportaciones que se han realizado desde el ámbito de los sistemas de los multicomputadores, respecto a la reutilización de técnicas de optimización o encaminamiento, gracias a la similitud que muestran ambos campos como, por ejemplo, la comunicación por mensaje o la demanda creciete de ancho de banda.

Así mismo, se introdujeron diferentes alternativas a esta tecnología, como son Frame Relay o X.25, realizando una comparativa que permitiera observar las ventajas que ofrece ATM frente a las anteriores tecnologías, especialmente en aquellas redes donde el tipo de tráfico que soporta es muy diverso y con una calidad de servicio (QoS) aceptable, minimizando tanto la complejidad de conmutación como la capacidad de proceso.

Se presentó el modelo de referencia del protocolo ATM con las principales funciones de las capas que lo componen, es decir, tanto la descripción de la capa física, la capa ATM, además de la capa de adaptación, así como las diferentes técnicas de control del tráfico y de la congestión en ATM. Dichas técnicas contemplan el control de la admisión de la conexión y de los parámetros de uso, mantenimiento de los recursos de la red, suavizado del tráfico y control de propiedades. Se estudiaron a su vez el modelo arquitectónico ATM, definiendo todos los parámetros que determinan su nivel de prestaciones, así como los diferentes tipos de conmutadores ATM, clasificados de acuerdo a las diferentes técnicas de conmutación por división de frecuencia o por división de tiempo.

El curso finalizó con la introducción de diferentes técnicas de mantenimiento de tráfico en redes ATM, que aseguren a la red y al usuario frente a pérdidas acusadas de la calidad de servicio (AoS). Se trata de técnicas cuya selección se basa en criterios de escalabilidad, optimización y robustez, y que conforman las particularidades de este tipo de redes.

2.3.6. Redes de interconexión.

Este curso fue impartido por el Dr. José Duato Marín. Su objetivo fue, por un lado, analizar los requisitos de las redes de inerconexión, presentando los aspectos más importantes del estado del arte de éstas, y por otro lado, evaluar las prestaciones de distintas alternativas de diseño en este entorno, proponiendo posteriormente posibles vias para la mejora de dichas prestaciones.

El curso está dedicado a profundizar en distintos aspectos de diseño de las redes de interconexión usadas en multiprocesadores y multicomputadores, presentando y evaluando las distintas alternativas propuestas. Antes, se recuerda lo más esencial de las arquitecturas de este entorno: espacio de direcciones de memoria, estructura de los nudos de conmutación, escalabilidad de la red, sincronización entre nodos, etc.

En lo referente a técnicas de conmutación y control de flujo, se presentan técnicas como Wormhole, Virtual Cut-Through, Mad Postman, conmutación de circuitos segmentada, Scouting o Canales Virtuales. Se analizan las ventajas e inconvenientes de las distintas técnicas mediante criterios definidos, como puede ser el cálculo de la latencia de un mensaje para cada una de ellas.

En cuanto a las distintas aproximaciones a la topología de la red, se estudian las redes multietapa clásicas y bidireccionales, toros, mallas y redes directas, relacionando estas aproximaciones con las técnicas de conmutación estudiadas anteriormente, de manera que pueda establecerse una correspondencia entre cada topología su técnica de conmutación óptima.

Un punto especialmente relevante del curso es el relativo al encaminamineto en este tipo de redes de interconexión. En este caso se estudian distintos tipos de algoritmos de encaminamiento (distribuidos, basados en información local, basados en máquinas de estados finitos, deterministas frente adaptativos, etc.). De gran interés resulta el estudio de aquellos algoritmos de encaminamiento que garantizan la libertad de bloqueos (DeadLocks) en la red. Para facilitar el diseño de tales algoritmos se introduce la herramienta del grafo de dependencias entre canales, y se establecen las condiciones que deben darse para que en la red exista plena garantía de libertad de bloqueos durante el encaminamiento.

Por último, se ofrece una panorámica de los nuevos desarrollos que se realizan o están previstos en distintos aspectos de las redes de interconexión de multicomputadores.

2.3.7. Programación lógica avanzada.

El curso, impartido por la Dra. María Alpuente Frasnedo, Mª José Ramírez Quintana y Germán Vidal Oriola. Tuvo como objetivo principal analizar cómo funciona la programación lógica y funcional a todos los niveles, tanto a nivel de desarrollo como internamente el interprete.

El curso comenzó con una visión general de las lógicas modales, donde se estudiaron los distintos tipos de lógicas modales: temporales, dinámicas y Epistémicas, así como se examinaron los distintos axiomas que pueden intervenir en cada tipo de lógica y la conversión a programación lógica clásica.

En la siguiente parte, se mostraron técnicas de mejora y especialización de los programas lógicos y funcionales, se vió como repercute en la eficiencia de los programas la forma de realizar algunas tareas, viendo conceptos como evaluación perezosa, delplegado, etc...

En la tercera parte se vieron conceptos de Programación Lógica Visual, realizando algunos ejemplos de programas sobre lenguajes visuales como: VLP, Pictorial Logic Programming, CUBE y VisualLogic.

Finalmente, se vio cómo funciona internamente el interprete de PROLOG (en que orden se van ejecutando los distintos predicados, cómo gestiona la memoria el compilador de PROLOG - viendo cómo realiza las tareas apilado de llamadas recursivas, gestión de parámetros, etc...), así como utilizar esos conocimientos para realizar programas.

2.3.8. Técnologia Software Orientada a Objetos.

El curso, impartido por la Dr. Isidro Ramos Salavert y Javier Oliver Villarroya tuvo como objetivo principal estudiar desde una perspectiva más formal el funcionamiento de la Programación Orientada a Objetos.

En primer lugar se repaso el funcionamiento de los Lenguajes Orientados a Objetos, repasando los elementos fundamentales de este paradigma, así como los distintas maneras de realizar acciones básicas para la Programación Orientada a Objetos. También se vieron distintas propuestas para especificaciones en algunos lenguajes de programación, también se vieron algunas propuestas de representación para los sistemas de objetos y el uso de objetos para algunas aplicaciones.

2.4. Conclusiones.

Para finalizar este apartado sobre las asignaturas realizadas se presentan las conclusiones obtenidas. Se dividen en dos bloques:

- Asignaturas cursadas durante el curso 2010/2011: Completé mi formación relacionada con la lógica difusa durante el curso *Programación Internet con lenguajes declarativos multiparadigma*, y por otro lado, asistí al curso *Técnologia Software Orientado a Objetos* en el que actualicé mis conocimientos en cuanto a las técnicas actuales de desarrollo de software, así como los relativos a los métodos actuales de creación de interfaces.
- Asignaturas que realizé en el año 1998 y que me han sido convalidado: debido al tiempo transcurrido desde el momento de su realización hasta la fecha no están directamente relacionados con el objeto de mi trabajo fin de Master.

3

Estado del Arte

En esta sección se van a estudiar los métodos de representación y comparación de series de datos temporales. En la primera parte, se presentan las distintas técnicas de representación de series temporales que se encuentran en la bibliografía. En la segunda parte, se muestran las distintas técnicas de comparación de series, así como los refinamientos y mejoras sobre los mismos.

3.1. Representación e indexación de series temporales.

La representación de los datos de un sistema es uno de los puntos más importantes, ya que la representación seleccionada influye en los procesos sobre los datos, por ejemplo, en el proceso de búsqueda. A continuación se van a exponer las técnicas de representación e indexación más importantes que se han encontrado en la bibliografía.

3.1.1. Resampleo.

En 1969 Astrom [47] creó el método más sencillo que consiste en tomar un valor de muestra cada cierto espacio de tiempo. Es una técnica fácil de implementar, pero tiene el inconveniente de que modifica la forma de la gráfica.

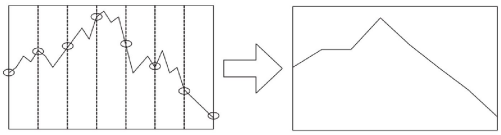

Figura 3.1: Ejemplo de reducción por muestreo.

La Figura 3.1 muestra un ejemplo de cómo funciona este método. En la parte izquierda se ve la gráfica de la forma real de la serie que consta de 8 picos y 8 valles, mientras que en la parte derecha se muestra la representación mediante este método, con tan sólo 2 picos y 1 valle.

3.1.2. Aproximación global a trozos (PAA).

Keogh et al. [53] realizaron una mejora en el año 2000. Ésta consiste en tomar como valor la media de todos los valores de un intervalo. A esta técnica se denominá Aproximación global a trozos (PAA – Piecewise Aggregate Approximation).

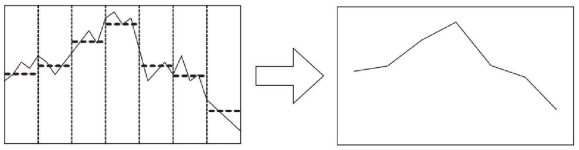

Figura 3.2: Técnica PAA(Piecewise Aggregate Approximation).

La Figura 3.2 muestra un ejemplo de esta técnica. En la gráfica de la izquierda se aprecia cómo cada intervalo se ajusta a un valor mediante la Ecuación 3.1.

$$\hat{p_k} = \frac{1}{e_k - s_k + 1} \sum_{i=s_k}^{e_k} p_i \tag{3.1}$$

donde s_k y e_k denotan los puntos de comienzo y fin de cada segmento k.

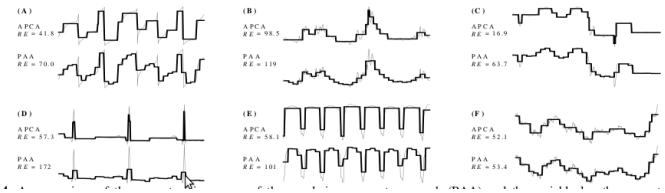

Figura 3.3: Técnica APCA.

3.1.3. Aproximación constante de adaptación a trozos (APCA).

Keogh [52] mejoró el metodo anterior presentando lo que denominó aproximación constante de adaptación a trozos (APCA – *Adaptive Piecewise Constant Approximation*). Como novedad permite que la longitud de los segmentos sea variable. Ello hace que la forma del resultado pueda ser modelada. La Figura 3.3 muestra un ejemplo de uso de esta técnica. En esta Figura se aprecia cómo la forma de las señales modeladas mediante APCA es mucho más cuadrada que las modeladas mediante PAA (Sección 3.1.2).

3.1.4. Compresión de características

La necesidad de comprimir el tamaño de los datos hizo que en un primer momento se recurriera a técnicas matemáticas. Surgieron dos técnicas.

En primer lugar, surgieron métodos basados en la transformada de Fourier (DFT - Discrete Fourier Transform). En 1990 Beckmann [71] propone utilizar los coeficientes DFT en una estructura arbolescente indexada en *R*-Tree*. En fase de post-proceso se realizan los descartes de incorrectos, sugiere utilizar la distancia euclídea para esto.

Muchos documentos han ampliado este enfoque, por ejemplo, para manejar escalamiento y las diferencias [77], subsecuencia correspondiente utilizando rectángulos delimitadores mínimos ([22], [102]), la formalización de restricciones de consulta e incorporarlas a la indexación procedimiento ([38], [78]), utilizando los últimos k coeficientes de DFT con la propiedad conjugado de la DFT [79], o el uso de Haar DWT en lugar de la DFT [17].

En segundo lugar, como mejora de la anterior técnica surgió la compresión Transformada Discreta de Wavelet (DWT - *Discrete Wavelet Transform*). Este tipo de compresión lo que hace es guardar la misma información con distintos tamaños lo que permite tener varias resoluciones. El origen de esta técnica se puede considerar en 1895, gracias a un trabajo de Karl Weierstrass [51], aunque fue Haar [3] quien en 1910 crea el primer sistema ortogonal. Esta técnica posteriormente se ha utilizado para comprimir todo tipo de datos.

Morchen [69] creó una nueva técnica de compresión utilizando las dos técnicas analizadas. Se sabe que los valores de una serie temporal obedecen a un patrón. De ahí que, una de las primeras técnicas de reducción es eliminar las redundancias y los posible valores erróneos que contengan. Para ello se pueden utilizar cualquiera de los dos métodos vistos anteriormente.

Para mantener la calidad óptima ambos métodos deben extraer los coeficientes más largos guardando las posiciones de los mismos. Para cada serie de datos se debe repetir este proceso. Esto produce una sobrecarga de memoria, y complica el proceso de comparación de elementos. Por este motivo, una solución al problema pasa por aplicar un recorte al número de coeficientes que se seleccionan - con lo que se disminuye el espacio de memoria requerido - y utilizar el mismo conjunto de coeficientes para todas las series del conjunto - lo que permite simplificar la tarea de comparación de series.

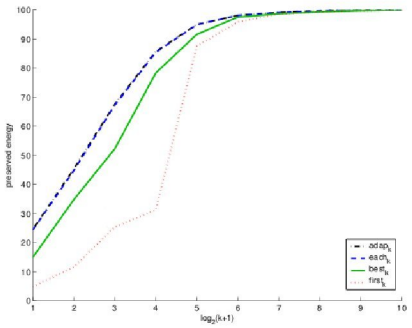

Figura 3.4: Coeficientes con DFT.

Las Figuras 3.4 y 3.5 muestran unas gráficas donde se observa, para cuatro funciones

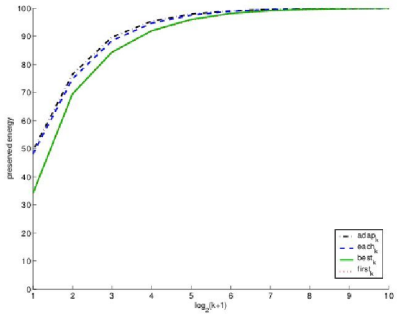

Figura 3.5: Coeficientes con DWT.

de distancia, la pérdida de información en función del número de coeficientes. Además de decidir el número de coeficientes, también es necesario examinar cómo se almacenan en disco, para que sea más eficiente.

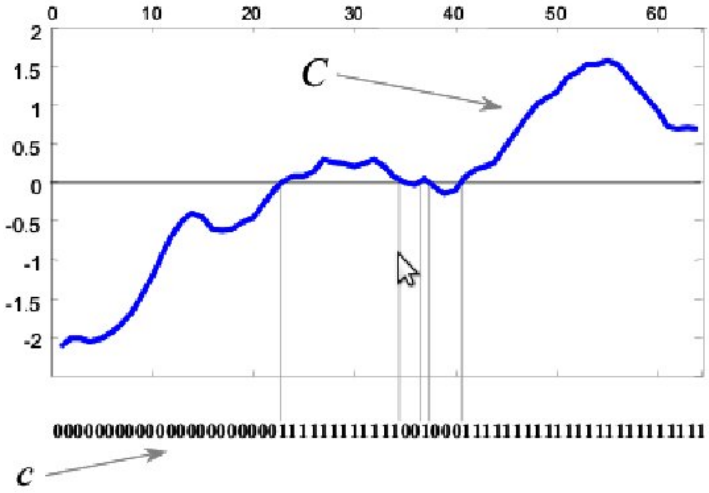

Figura 3.6: Representación mediante un bit.

3.1.5. Representación mediante un bit.

Fue presentada por Ratanamahatana et al. [82] en 2005. La idea es convertir cada valor de la gráfica en un único dígito binario, convirtiendo a 0 los valores inferiores al umbral y a 1 los valores superiores. Para obtener el valor del umbral se utiliza la media de los valores de la serie (Ecuación 3.2). La Figura 3.6 muestra un ejemplo.

$$f(n) = \left\{ \begin{array}{ll} 0 & \textbf{si } n > umbral \\ 1 & \textbf{si } n < umbral \end{array} \right\} \tag{3.2}$$

La señal de entrada queda representada como una sucesión de ceros y unos (Figura 3.6). La serie se representa como *22,0,11,2,1,33,...*, lo que significara que hay 22 ceros, 11 unos, 2 ceros, 1 uno, etc. Esto reduce considerablemente la información respecto a los datos originales.

3.1.6. Aproximación a lineas rectas (PLR - Piecewiese Linear Representation).

Keogh [54] propone cómo crear un reducción importante del tamaño de los datos con un método que no está basado en aproximaciones de Fourier, siendo además muy versatil y eficiente. A continuación se explica cómo funciona este método.

En el primer paso se realiza una aproximación creando grupos de 3 valores, ya que el objetivo es que ningún segmento tenga un número inferior a 3 valores. El último segmento puede contener entre 3 y 5 valores. Para cada segmento se busca la mejor aproximación a una recta usando la ecuación de regresión clásica (Ecuación 3.3).

$$y - \bar{y} = \frac{s_{xy}}{{s_x}^2}(x - \bar{x}) \tag{3.3}$$

Los puntos no tienen por qué estar perfectamente alineados. Por ello, el error normalizado que se comete se calcula mediante la Ecuación 3.4.

$$e_i = \frac{\sum_{m=1}^{j} d_m^2}{j} \tag{3.4}$$

A continuación, se va realizando la mezcla de cada pareja de segmentos s_i con s_{i+1} generando una nueva aproximación de la linea. Este proceso itera hasta que existe una sola aproximación a la linea. Los detalles de este método se pueden ver en [54].

3.1.7. Puntos de importancia Porcentual (PIP - Perceptualy points).

Creado en 2001 por Chung [26] y, posteriormente usado para aplicaciones financieras, por ejemplo, por Fu [33] en 2008.

Los datos de entrada son una serie de valores que se denominarán P, y están formados por p_1,p_2,...,p_n, siendo n la longitud de la serie de datos.

En primer lugar, se ejecuta el proceso de identificación de puntos de interés que considera que el primer y el último valor forman parte de la solución. A continuación, se toma el punto más lejano, y se une con la solución inicial. Este bucle se va repitiendo hasta que se tengan todos los valores recesarios.

La Figura 3.7 muestra un ejemplo donde se han seleccionado siete valores de todos los posibles.

Ahora falta por definir la función a utilizar para calcular la distancia entre dos puntos. Fu [33] propone tres alternativas: (1) la distancia euclídea; (2) distancia perpendicular y, (3) distancia vertical.

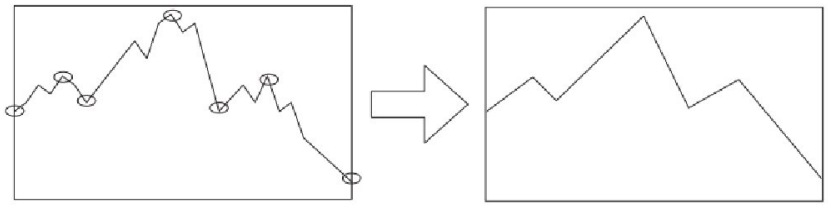

Figura 3.7: Técnica PIP.

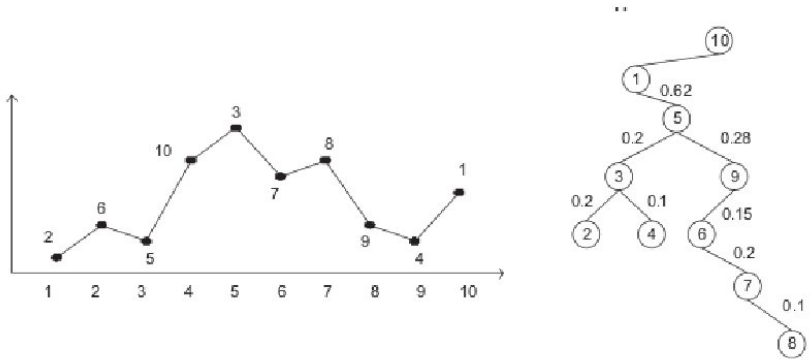

Figura 3.8: Ejemplo de representación con PIP.

También sugiere una representación en arbol para poder almacenar los valores de una manera más eficiente. La estructura es denominada arbol binario especializado (Figura 3.8). Esta Figura muestra cómo queda el árbol a partir de unos datos de entrada. En el primer paso, se eligen 10 valores (parte izquierda), y cuando se almacena en memoria, se crea un árbol como el de la parte de la derecha.

Sobre esta técnica Pratt y Fink [72] propusieron mejoras donde fijaron unos límites máximos y mínimos a la hora de la obtención de los puntos de interés. Fink [32] en 2003 propuso una técnica para su indexación.

Pratt y Fink fijan una ratio de reducción para saber el número de valores que se deben almacenar de la serie de valores. Respecto a la distancia entre los puntos consideran tres posibles formulas: (1) diferencia media (Ecuación 3.5); (2) diferencia mínima (Ecuación 3.6) y; (3) mínimos cuadrados (Ecuación 3.7).

$$\frac{1}{n} * \sum_{i=1}^{n} |a_i - b_i| \tag{3.5}$$

$$max_{i \in [1..n]} |a_i - b_i| \tag{3.6}$$

$$\sqrt{\frac{1}{n} * \sum_{i=1}^{n} e(a_i - b_i)^2} \tag{3.7}$$

3.1.8. Métodos basados en polarización.

Es una técnica pensada para poder analizar series de datos que van llegando de forma continua (streaming). La idea consiste en utilizar un almacenamiento más en detalle para datos actuales, mientras que se usa una representación de menos detalle para los datos menos actuales. Para conseguir este fin, se comprimen los datos más actuales con mejor calidad, y los datos más antiguos con peor calidad. Este método se divide en dos tipos: segmentos iguales y segmentos de longitud variable.

A continuación se muestra la notación utilizada durante todo este apartado:

S: serie temporal.

N: número de puntos de la serie S.

K: número total de coeficientes que serán guardados de la serie.

m: número de segmentos.

s_i: cada segmento de la serie reducida.

n: longitud del segmento.

n_i: longitud del segmento cuando sea variable.

k: número de coeficientes guardados de cada segmento de longitud variable.

k_i: número de coeficientes guardados para el segmento en el caso de longitud iguales.

Para realizar la compresión de cada segmento se puede utilizar cualquiera de las técnicas examinadas anteriormente. Como los métodos más usados son DFT y DWT, y estos métodos funcionan mejor con series que sean potencia de dos, se busca que la longitud de los segmentos tengan esa característica. A continuación se detallan las dos técnicas.

A) Segmentos con longitud igual

Fue desarrollada por Zhao [105] en 2006. La idea es comprimir los datos más antiguos con calidad inferior, con lo que ocupan menos espacio, y los datos más recientes con mejor calidad. Cada segmento s_i es más próximo en el tiempo cuanto menor es el subíndice, lo que hace que para calcular el valor de k_i se deba tener presente una función que decremente el valor de k_i. A continuación se citan dos ejemplos: (1) una función decreciente monótona lineal (Ecuación 3.8) y; (2) una función decreciente exponencial (Ecuación 3.9).

$$k_i = \begin{cases} p - i & \text{si } p > \text{i}, \\ 0 & \text{en caso contrario,} \end{cases} \tag{3.8}$$

$$k_i = \begin{cases} \frac{p}{2^i} & \text{si } p \geq 2^i, \\ 0 & \text{en caso contrario,} \end{cases} \tag{3.9}$$

Estas técnicas son efectivas en los casos en que se estén recibiendo datos contínuamente. Si el número de datos recibidos sin introducir en ningún segmento no excede el tamaño máximo del segmento mientras se van recibiendo datos, se van poniendo a continuación de los segmentos. Cuando el número de datos alcanza a n se crea un nuevo segmento llamado s_1, y se cambia el nombre de todos los demás segmentos de s_i a s_{i+1}. Para una explicación más detallada sobre el funcionamiento de este método ver [105].

B) Segmentos con longitud diferente.

Existe una variación que consiste en no fijar el tamaño de los segmentos. Con esto se busca que los segmentos con información más actuales se compriman con mejor calidad y más puntos, mientras que los más antigüos que tienen una calidad de compresión menor. Existen varios criterios a la hora de tomar el tamaño de la ventana de actuación que se debe utilizar. Yixin Chen [101] propone un método donde el tamaño de la ventana viene controlada por el tiempo y el calendario. También surgieron técnicas basadas en un crecimiento exponencial [11] y un plazo piramidal que también está compuesto de una parte exponencial [6].

En el esquema que propone Chan los tamaños de los segmentos están fijados como potencias de 2 para mejorar la gestión del espacio. El funcionamiento más en detalle de esta técnica se muestra en Zhao [105].

Tanto para las técnicas de longitud fija como variable sólo debe cambiar un único segmento cuando llega un nuevo dato. El resto se quede como estaba, lo que supone una ganancia en eficiencia.

3.1.9. Suma de variación de segmentos (SSV).

Esta técnica fue propuesta por Lee [62] en 2003, y mejora la calidad de los datos para su posterior consulta. El objetivo de la suma segmentada de indexación de la variación (*SSV-indexación*) es extraer características sobre la secuencia que aporten información. Esto crea la necesidad de definir una función de distancia que satisfazga la condición de límite inferior de la distancia mínima. Sea un conjunto de secuencias de tiempo de longitud n, la idea del método consta de dos pasos:

1. Dividir cada secuencia de tiempo en segmentos de igual longitud l. El punto de inicio y final de los segmentos adyacentes deben ser iguales, es decir, para cada segmento s_i, su punto final será el mismo que el punto inicial de s_{i+1}.

2. Extrae una función simple de cada segmento. Se propone utilizar la suma de la variación de la función de un segmento en una secuencia temporal. FA_j denota la característica del segmento j-ésimo de una secuencia de tiempo A. Se define un vector de características de una secuencia de tiempo A como $FA =< FA_1, FA_2, \ldots FA_n >$ donde cada FA_j se obtiene mediante la Ecuación 3.10.

$$FA_s = \sum_{i=(s-1)l+(2+s)1}^{s(l-1)} |a_{i+1} - a_i| \tag{3.10}$$

La Figura 3.9 ilustra la técnica de reducción de dimensionalidad. Una secuencia de tiempo de longitud 13 se proyecta en tres dimensiones. La secuencia de tiempo se divide en tres segmentos y se obtiene la suma de la variación de cada segmento. En el ejemplo, si la serie original está formada por los valores (5,4,5,6,8,7,7,5,4,3,4,5,7), el resultado sería FA=(SSV(5,4,5,6,8),SSV(8,7,7,5,4),SSV(4,3,4,5,7)).

Demuestran que el límite inferior de la distancia mínima entre los vectores de características es una condición que garantiza que no habrá descartes incorrectos a la hora de realizar una búsqueda. Más detalles en [62].

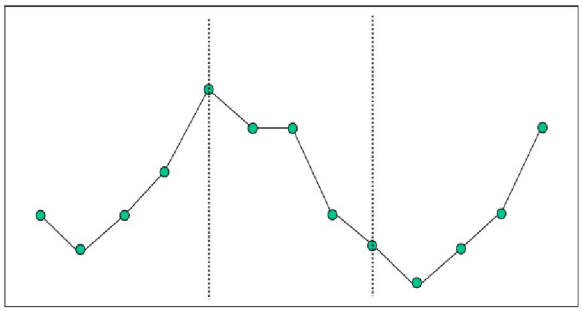

Figura 3.9: Ejemplo de reducción SSV.

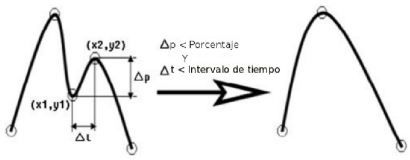

Figura 3.10: Técnica CPM.

3.1.10. Modelo de puntos críticos (CPM).

Bao [7] propone un método que utiliza el concepto de puntos críticos, que son puntos máximos y mínimos dentro de un intervalo local. De todos los puntos de la serie, se deben descartar los que no se utilizarán, proponiendo después una condición que debe cumplir cada punto que no se vaya a eliminar. La Figura 3.10 muestra un ejemplo de cómo se toman los puntos.

Para decidir los puntos que forman parte de la representación se define la Ecuación 3.11 que indica las pendientes que se deben mantener.

$$X_{i+1} - X_i < T \; and \; \frac{y_{i+1} - y_i}{(y_i + y_{i+1})/2} < P \tag{3.11}$$

3.1.11. Basados en conjuntos difusos.

La primera propuesta la realizó Song and Chissom [74] en 1993. En esta publicación toman dos ejemplos habituales de series temporales (predicciones temporales y el estado de anímico de las personas) y plantean un marco de trabajo común. En los dos casos se puede apreciar como para indicar los conceptos es más claro y fácil utilizar un literal (fatal, mal, regular, bien, muy bien,...) que un número. Además, define dos tipos de series de datos temporales: variantes (si la serie es infinita) e invariantes (si la es finita). En trabajos posteriores([73], [75]) propone 3 métodos distintos, concluyendo que el que mejor resultado da es el basado en redes neuronales. En 1996, Song y Chissom [76] continuaron trabajando en esa dirección y compararon sus modelos con otras técnicas basadas en Modelos de Markov.

Para convertir los datos temporales en datos difusos se deben realizar dos pasos. El primer paso se llama fuzzificación y el segundo debe guardar las reglas para que opere el sistema.

3.1.11.1. Fuzzificación.

Han habido varios métodos propuestos para realizar este proceso con series de temporales. El primer método que se utilizó fue la discretización basada en la partición del universo de discurso, y posteriormente se propuso otro mediante agrupamiento.

La idea del primer método, es dividir el universo del discurso en intervalos (iguales o desiguales) para la posterior unificación de los valores de cada intervalo en un único valor difuso. Varios autores han propuesto métodos relacionados, a continuación se ven algunos ejemplos:

- Song and Chissom.- En 1993 crearon una propuesta de representación. En [73] y [75] propusieron mejoras concluyendo que el mejor es el basado en redes neuronales.

Algoritmo 1 Método de Chen

1. Partir el universo de discusión en intervalor de longitud igual.
2. Definir los conjuntos difusos de mi universo.
3. Fusificar los datos historicos.
4. Identificar las relaciones difusas.
5. Establecer las relaciones entre grupos.
6. Defusificar la salida prevista.

- Chen.- En 1996 Chen [19] detecta el inconveniente de la velocidad de cálculo en las técnicas propuestas por Song y Chissom debido al operar con matrices y determinantes. Con el fin de reducir el proceso de cálculo propone un nuevo modelo (Algoritmo 2) que es más eficiente al eliminar operaciones de cálculo, y además, es más robusto.
- Tsaur.- En 2005 [83] utilizó el concepto de entropía para determinar el valor mínimo de tiempo de un índice t para minimizar el error.
- Singh.- En [90] propone un método basado en el uso de palabras.

Un parámetro importante es la longitud de intervalo. El efecto de éste fue estudiado en 2001 por Huarng [50] planteando dos métodos: uno basado en la media, y otro, en la distribución. Posteriormente en 2006 propuso en [43] un algoritmo donde la longitud del intervalo es dinámico, el cual sería revisado por otros autores como: Yolcu [95], Davari [85], Kuo [45, 46], Park [49], Hsu [40], Fu [35] y Huang [41] que utilizó una técnica de optimización basado en una nube de puntos para determinar la longitud del intervalo dinámico. Lee [64, 65] también emplea un algoritmo genético para el mismo propósito.

El segundo método que se utilizó consistió en realizar agrupaciones de los valores más próximos en grupos. Existen dos formas de realizar este agrupamiento:

1. C-media.- utilizada por Cheng [14] y Li [92]. Tiene en cuenta la distribución de los datos y la incertidumbre de los mismos, asignando un grado de pertenencia de grupo a cada grupo. El objetivo es minimizar el valor de la Ecuación 3.12.

$$f_w(\gamma, M) = \sum_{i=1}^{n}\sum_{j=1}^{c}(\gamma_{ij}^{a})||x_i - m_j||^2 \tag{3.12}$$

donde x_i es el i-ésimo elemento de un conjunto de datos $\{x_1, x_2, ..., x_n\}$, $x_i \in$ R. c es el número de grupos, c $\in \{2, 3, ..., n-1\}$. w es una constante ponderada y a $\in$ (1, ∞). $y = [\gamma_{ij}]$, donde γ_{ij} es el grado de pertenencia de x_i perteneciente al grupo j. M = $\{m_1, m_2, ..., m_c\}$, donde m_j es el centro de j clúster. $\|*\|$ es la medida de similitud entre x_i y m_j.

2. Jerárquica.- Lee en [103] propone un sistema jerárquico que utiliza un algoritmo de agrupamiento jerárquico. Consta de dos niveles: Capa superior que se encarga de los grupos, y capa inferior que se encarga de los conjuntos difusos. El algoritmo de correlación cruzada que se encarga de actuar en la parte superior con los grupos, y el algoritmo c-media que se encarga de actual sobre los conjuntos difusos.

En [20] Chun-Hao Chen proponen un método básado en ventanas que utiliza parámetros como: valores de una serie S, conjunto de valores y el tamaño de ventana. Crea como resultado un conjunto de relaciones. En la parte del algoritmo de fuzzificación utiliza conjuntos difusos para convertir cada valor numérico a valor difuso (Figura 3.11).

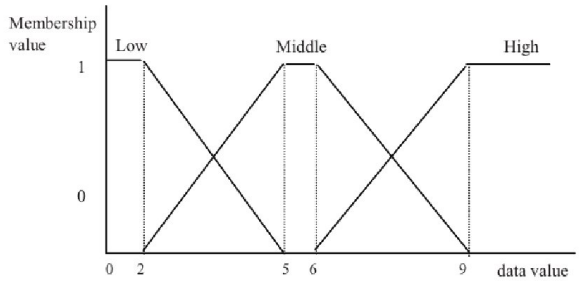

Figura 3.11: Conjuntos Difusos.

3.1.11.2. Relaciones difusas.

Al trabajar con series de tiempo se deben almacenar las relaciones que existen entre los distintos conjuntos difusos que intervienen en el sistema. A continuación pasamos a ver las distintas soluciones que se han ido dando a este problema.

Sullivan y Woodall [1] usan una matriz de transición sobre la base de una cadena de Markov en lugar de utilizar una matriz para la relación lógica. Chen [19] propone un método que sustituye la matriz de transición por un conjunto de tablas. Esta misma idea es utilizada por otros autores (Huarng [50], Yu [104], Huarng and Yu [43], Cheng et al. [15], y Egrioglu et al. [30]. Huarng y Yu [42] sugiere un método basado en redes neuronales para gestionar las relaciones que está siendo utilizado por muchos autores. Aladag et al. en [12] establece las relaciones difusas mediante redes neuronales de avance. Posteriormente en [13] emplean redes neuronales recurrentes Elman. Yu y Huarng [44], y Yolcu et al. [94] propusieron un enfoque diferente en el que los valores de función de pertenencia se emplean para utilizar

redes neurales de alimentación de avance en la fase de determinación de las relaciones difusas.

Algoritmo 2 Algoritmo de reglas difusas

```
R_1 = { reglas de primer orden } .
for d = 2 → R_{d-1} ≠ ∅ do
  T = { cada conjunto antecedente de l|l ∈ R_{d-1} } .
  for all αelementosdeT do
    for all C_q, q ∈ {1, ..., m} do
      calcular∅_{F_q,C_q}
      if interesa(∅_{F_q,C_q}) then
        R_d = R_d ∪ GenerarRegla(F_q, C_q)
      end if
    end for
  end for
end for
```

En 2004 Wai-Ho y Keith Chan [99] proponen una técnica que es resistente a ruidos. Para distinguir las asociaciones más interesantes utiliza el análisis ajustado residual, que tiene la ventaja de no tener definidos umbrales por el usuario. El Algoritmo 2 se utiliza para crear estas reglas. Este algoritmo utiliza el concepto de orden en sus reglas, según este orden, la regla de primer orden es la que contiene en el antecedente un conjunto difuso, la de segundo orden es la que dos, y si se generaliza la expresión una regla de orden n es la que contiene n conjuntos difusos.

En 2012 Chun [20] propone un método basado en ventanas. El algoritmo genera las relaciones mediante tablas, aportando como novedad un resultado más cercano al lenguaje natural. A continuación se muestra un ejemplo. Si se supone que el valor de confianza mínimo ha sido fijado en 0.65, tras realizar todos los cálculos de algoritmo la regla *Si A1 = Medio entonces A4 = Medio* tiene un factor de confianza asociado de 0.72. Esto se puede traducir con la expresión *Si hay un valor medio, con mucha probabilidad, el valor de tres instantes después va a ser Medio también.*

3.2. Búsqueda de patrones.

Ahora se pasa a exponer las distintas maneras de comparación de series de datos. Algunos métodos de representación vistos comparten la técnica de comparación de valores. Tradicionalmente, el criterio utilizado cuando se comparan dos valores en las bases de datos es la exactitud. Esto significa que el resultado de la comparación es '*sí*' cuando se cumple la condición o '*no*' en caso contrario. En series temporales no se puede realizar la comparación con ese criterio debido a que cuando se comparan dos series temporales es muy difícil que sean exactamente iguales. Se suele cuantificar el grado de *igualdad/desigualdad* que tienen los dos elementos de la comparación pudiendo así cuantificar la calidad del resultado de la comparación.

Dado que existe una gran dependencia entre la función de similitud y el significado de los datos es difícil crear una función de similitud genérica adecuada. En bases de datos que contienen información de series temporales se pueden obtener dos tipos de resultados en función de la longitud de los elementos comparados: (1) cuando la longitud de los dos

elementos son iguales, el resultado esta formado por un número para cada serie de la base de datos que indicará el grado de similitud/diferencia que existe; (2) cuando la longitud del elemento de la base de datos es mayor que los datos de la consulta. Formalmente el resultado será una tupla $\{P_i, V_i\}$ para cada comparación, donde P_i es la posición en la que se encuentra y V_i es el grado de similitud.

Una vez vistas las generalidades de la función de similitud, se detallarán las distintas soluciones que han ido aplicando los distintos autores en relación con la función de similitud.

3.2.1. Funciones Matemáticas.

En este apartado se va a detallar todos los métodos de realizar la comparación que están basados en procedientos matemáticos.

En 1993, Agrawal [5] propone el uso de la función de mínimos cuadrados (Ecuación 3.13), mientras que, para el almacenamiento usa la transformada de Fourier. El método de selección por el que se decanta Agrawal está basado en la distancia euclídea, entiende que es el óptimo [2]. Para realizar la búsqueda parcial de subsecuencias dentro de la secuencia, localiza el mayor trozo que mejor enlaza con el elemento a buscar, y retorna dos valores: el punto donde se encuentra el trozo y el resultado de la comparación.

$$D(\vec{x}, \vec{y}) \equiv \sqrt{\sum_{t=0}^{n-1} |x_t - y_t|^2} \tag{3.13}$$

En la literatura se pueden ver otros métodos además de la distancia euclídea. Goldin y Kanellakis [39] en 1995 extienden el trabajo de Agrawal [5] proponiendo una función de similitud basada en restricciones. Para conseguirlo formaliza una sintaxis que permite establecer distintos tipos de condiciones. Posteriormente en 1997, Das [27] muestra una función de comparación basada en propiedades geométricas, exponiendo los distintos casos que se pueden dar a la hora de comparar los distintos valores de las series temporales.

Bozcaya [10] modificó en 1997 el método de la distancia euclídea entendiendo que dos secuencias enlazan cuando la mayoría de los puntos enlazan. Para hacer las dos secuencias comparables lo primero que se debe hacer es añadir los puntos que falten para que ambas secuencias tengan la misma longitud en el momento de aplicar el criterio de comparación. La distancia entre cada punto no puede ser superior a un umbral determinado.

En 1998 Chu [24] muestra la problemática de utilizar la distancia euclídea. En la Figura 3.12 (en rojo la solución ofrecida por la distancia euclídea) se muestran los dos problemas encontró Chu: (1) Cuando una gráfica está a distinta altura que la otra, y (2) Cuando los dos elementos que se comparan tienen una forma similar pero contraída. Ante estos problemas propone una nueva función de similitud (Ecuación 3.14) que trata a cada serie como si fuera una sucesión de triángulos, y va comprobando que la pendiente del triángulo no expede un valor límite que establece el usuario.

$$\epsilon \leq (D_{i+1} - D_i) - ((Q_{j+1} - Q_j) \leq \epsilon \tag{3.14}$$

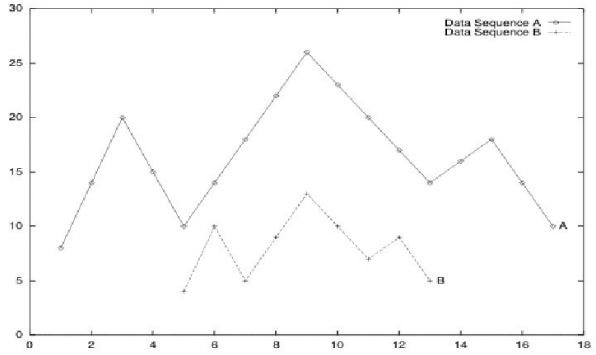

Figura 3.12: Problemática distancia euclídea. A) Distintas Altura. B) Datos contraídos.

donde ϵ es el umbral definido por el usuario, D_i son los distintos valores de la serie de la base de datos y Q_i son los distintos valores de la serie que se está tratando de evaluar.

En ese mismo año, Lam y Wong [60] proponen realizar un ajuste al método de comparación mediante la Ecuación 3.15. Este ajuste permite utilizar la distancia de Manchester ($k = 1$) o la distancia Euclídea ($k = 2$), no obstante, esto no es suficiente para poder calcular los mejores resultados. Por lo cual añade una nueva condición para controlar que la distancia entre dos puntos no puede exceda de un límite [60].

$$\sqrt[k]{\sum_{i=1}^{m} |A_i - B_i|^k} \leq \epsilon \tag{3.15}$$

En 2000, Gavrilov [36] utiliza el método de representación PAA (Sección 3.1.2) y estudia el mejor marco para la comparación de datos, llegando a la conclusión que la mejor medida de similitud es la distancia euclídea.

Chan [16] en 2003 plantea un sistema para el proceso del filtrado basado en la transformación de Wavelet Haar, el cambiar el método de reducción hace que también haya que modificar la técnica de normalización. El preprocesado de esta técnica consta de dos pasos:

- Elección del módelo de similitud: entre un módelo basado en la distancia euclídea (Ecuación 3.16) o uno basado en cambio de variable V (Ecuación 3.17).

$$D(\vec{x}, \vec{y}) = \sqrt{\left(\sum_{i=1}^{n-1} (y_i - x_i)^2\right)} \leq \epsilon \tag{3.16}$$

$$D(\vec{x}, \vec{y}) = \sqrt{\left(\sum_{i=1}^{n-1} ((y_i - x_i) - (y_A - x_A))^2\right)} \leq \epsilon \tag{3.17}$$

- Indexación de los elementos creados. Para calcular los índices, en primer lugar con una transformación de Haar y un factor de $1/\sqrt{2}$ se obtienen los puntos de la ventana ω. Los detalles de la estructura arbolescente se muestran en [16].

Una vez realizado el preproceso de los elementos de la consulta, se pasará a ejecutar la consulta en sí, mediante el criterio del vecino más próximo.

3.2.2. Distorsión dinámica (DTW - Distancia Dynamic Time Warping).

En primer lugar se va a ver la comparativa de distancia euclídea y DTW, a continuación verá el funcionamiento general del algoritmo y, finalmente, las distintas mejoras en las que han ido trabajando los distintos autores.

3.2.2.1. Comparación de Distancia euclídea y DTW

Uno de los problemas de la distancia euclídea es la alineación de datos. El algoritmo DTW corrige este problema. La Figura 3.13 ilustra el problema (parte superior) usando la distancia euclídea y cómo lo corrige DTW (parte inferior).

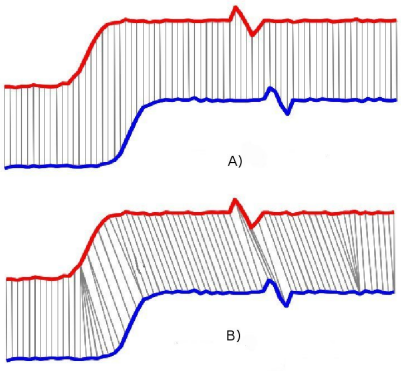

Figura 3.13: Comparación métodos: A) Distancia euclídea. B) DTW.

El algoritmo DTW tiene el inconveniente de tener una gran la complejidad. Esto hizo que al principio se utilizara más la distancia euclídea, pero posteriormente, con la mejora de la potencia de cálculo y la optimización del algoritmo, se haya extendido más esta técnica.

3.2.2.2. Algoritmo DTW

El algoritmo tiene dos parámetros de entrada: (1) Secuencia de datos Q (Ecuación 3.18), y (2) Cadena de búsqueda C (Ecuación 3.19).

$$Q = \{q_1, q_2, ...q_n\} \tag{3.18}$$

$$C = \{c_1, c_2, ...c_m\} \tag{3.19}$$

Para realizar la alineación se construye una matriz D de n x m elementos, donde cada elemento d_{ij} contiene la distancia (Ecuación 3.20) entre los elemento q_i y c_j . A partir de esta mátriz se define el camino de deformación ω. Éste es contínuo, y cada elemento K^{th} de W está definido como $w_k = (i,j)_k$ donde $W = w_1, w_2, \dots w_k$ y $max(m,n) \leq K < m + n - 1$. La Figura 3.14 muestra un ejemplo.

$$D(q_i, c_j) = (q_i - c_j)^2 \tag{3.20}$$

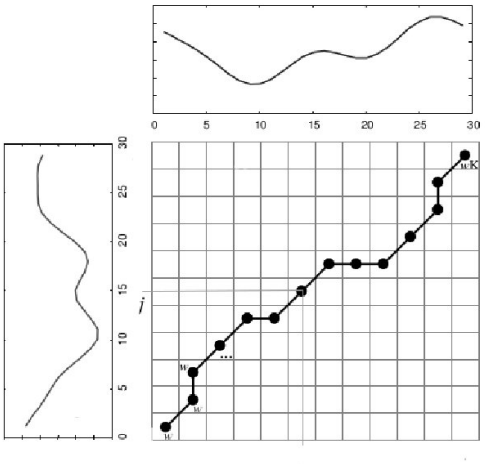

Figura 3.14: Un ejemplo del camino de deformación.

Este camino está marcado por las siguientes restricciones:

- Condiciones de contorno.- $w_1 = (1,1)$ y $w_k = (m,n)$ son el comienzo y final del camino, siendo k la longitud del camino de deformación.
- Continuidad.- Tomando un elemento del camino $w_i = (a,b)$ entonces $w_{i-1} = (a',b')$ donde $a \leq a'$ y $b \leq b'$. Es decir las celdas son adyacentes.
- Monotonía.-Tomando un elemento del camino $w_i = (a,b)$ entonces $w_{k-1} = (a',b')$ donde $a - a' \geq 0$ y $b - b' \geq 0$. Obliga a que los puntos de W estén espaciados en el tiempo de forma continua.

Hay varios caminos que satisfacen estas condiciones. La mejor solución va a ser la que cumpla la Ecuación 3.21.

$$DTW(Q,C) = min\sqrt{\sum_{k=1}^{K} w_k}/K \tag{3.21}$$

donde K es un coeficiente para ajustar los casos en que la longitud de cadena sea distinto.

3.2.2.3. Mejoras del algoritmo

Una vez explicado el funcionamiento básico de algoritmo, se va a ver las mejoras que se han ido realizando.

En 2000, Keogh y Pazzani [56] introducen una modificación para que se adapte a PAA (Sección 3.1.2). La Ecuación 3.22 se utiliza para la distancia. La Figura 3.15 ilustra como cambia la representación del método DWT (parte superior) y de PAA (parte inferior), se puede ver más detalles en [56].

$$d(\overline{Q_i}.\overline{C_j}) = (\overline{Q_i} - \overline{C_j})^2 \tag{3.22}$$

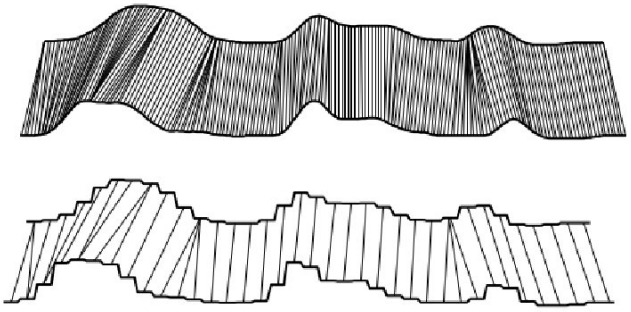

Figura 3.15: Uso de DWT con PAA.

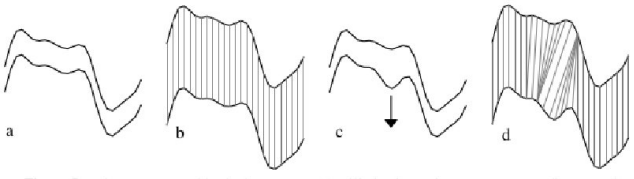

Figura 3.16: Usando DTW, dos secuencias identicas (a,b). Con un leve cambio en un valle (c,d).

Keogh y Pazzani [57] en 2001 mostrarán algunos problemas de DTW cuando las alineaciones de los datos son poco intuitivas (Figura 3.16), y también con alineaciones obvias cuando se buscan características simples (p.e., un pico, un valle, etc...). Este algoritmo denominado DDTW (*Derivative Dynamic Time Warping*) plantea una nueva forma de realizar la comparación (Ecuación 3.23), añadiendo una estimación que se calcula como la media de los puntos vecinos. La Figura 3.17 muestra la representación de los datos en los algoritmos DTW y DDTW.

$$D_x\left[q\right] = \frac{(q_i - q_{i-1}) + ((q_{i+1} - q_{i-1})/2)}{2} \tag{3.23}$$

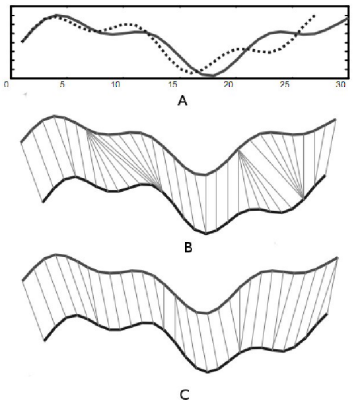

Figura 3.17: Representación de datos DDTW. A) Datos originales. B) Datos DTW. C) Datos DDTW.

Este mismo año Kim et al. [59] propone un método que consiste en guardar en la base de datos una tupla con los valores máximo, mínimo, mayor y menor. Creando un índice multiple (a través de una estructura arbolescente: R-tree, R*-tree o X-tree) con lo que reduce el espacio de búsqueda en el momento de ejecutar una consulta. La Figura 3.18 muestra un ejemplo de representación de los datos.

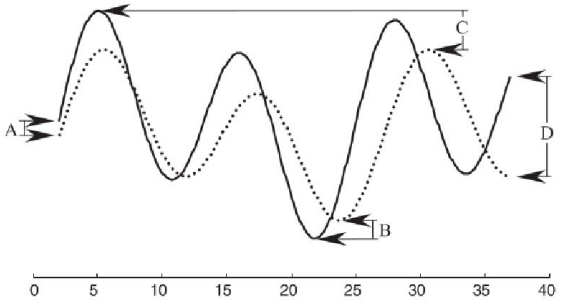

Figura 3.18: Parámetros del formato de Kim: Primero (A), Último (D), Mayor (C), Menor (B).

El problema de la velocidad de proceso fue aliviado mediante restricciones globales. Se puede ver una restricción global como un sistema para limitar los índices $w_k = (i,j)_k$ tal que $j - R_i \leq i \leq j + R_i$, donde R_i es un término que define el rango permitido de deformación, para un punto dado en una secuencia. Los tipos de restricciones más utitizados son: banda Sakoe-Chiba (Figura 3.19a) y el paralelogramo Itakura (Figura 3.19b). En el primer caso (Ecuación 3.24) la forma de calcular las celdas eliminadas depende de un R que es independiente de i; mientras que el segundo caso (Ecuación 3.25) R es una función de i.

$$R_i = \begin{cases} 5 & 1 \leq i \leq m-5 \\ m-i & m-5 < i \leq m \end{cases} \tag{3.24}$$

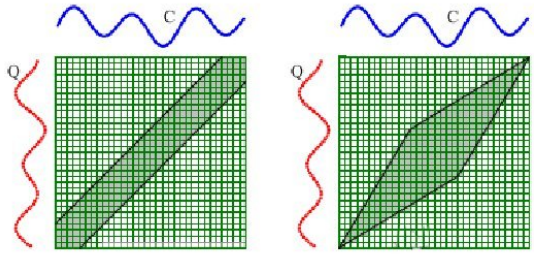

Figura 3.19: Restricciones más utilizadas: A) Banda Sakoe-Chiba. B) Paralelogramo Itakura.

$$R_i = \begin{cases} [\frac{2}{3}i] & 1 \leq i \leq [\frac{2}{3}i] \\ ([\frac{3}{8}m] - [\frac{2}{5}i]) & [\frac{3}{8}m] < i \leq m \end{cases} \tag{3.25}$$

Con el objetivo de indexar los datos Keogh [55] utiliza la idea de área de influencia. Este área depende de un valor r que define el grosor de la misma y que se combina con restricciones (Sakoe-Chiba y el paralelogramo Itakuta) y define dos series independientes: U (Ecuación 3.27) y L (Ecuación 3.26).

$$U_i = max(q_{i-r} : q_{i+r}) \tag{3.26}$$

$$L_i = min(q_{i-r} : q_{i+r}) \tag{3.27}$$

La Figura 3.20 ilustra cómo queda una secuencia al aplicarle el área de influencia. Los conceptos vistos se aplican sobre la versión del algoritmo de 2000 de Keogh, por lo que se van a explicar los cambios a realizar para poder ser indexado.

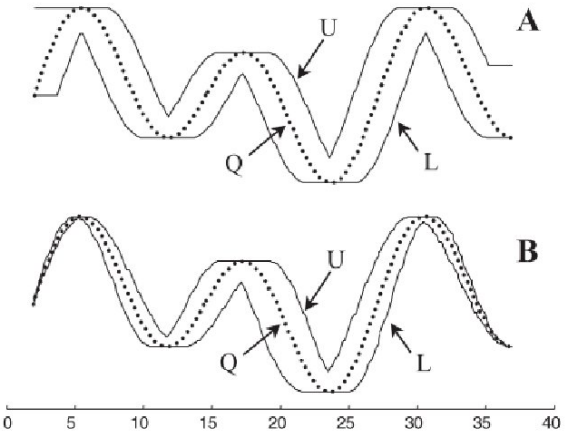

Figura 3.20: Secuencias U y L. A) Usando Sakoe-Chiba. B) Itakura.

$$LB\_PAA(Q,\overline{C}) = \sqrt{\sum_{i=1}^{N} \frac{n}{N} \begin{cases} (\bar{c}_i - \hat{U}_i)^2 & \text{si } \bar{c}_i > \hat{U}_i \\ (\bar{c}_i - \hat{L}_i)^2 & \text{si } \bar{c}_i < \hat{L}_i \\ 0 & \text{en caso contrario,} \end{cases}} \tag{3.28}$$

$$MINDIST(Q,\overline{C}) = \sqrt{\sum_{i=1}^{N} \frac{n}{N} \begin{cases} (\bar{l}_i - \hat{U}_i)^2 & \text{si } \bar{l}_i > \hat{U}_i \\ (\bar{ht}_i - \hat{L}_i)^2 & \text{si } \bar{ht}_i < \hat{L}_i \\ 0 & \text{en caso contrario,} \end{cases}} \tag{3.29}$$

En primer lugar, se deben marcar los límites de la representación, para lo que se crearán dos señales $U(Upper)$ y $L(Lower)$ y los tres elementos (consulta, L y U) se convierten a la representación PAA (Figura 3.21). Con todo lo visto anteriormente presenta el algoritmo [55] de búsqueda de vecino más próximo (K-NN), que es una optimización del Algoritmo GEMINI K-NN [31].

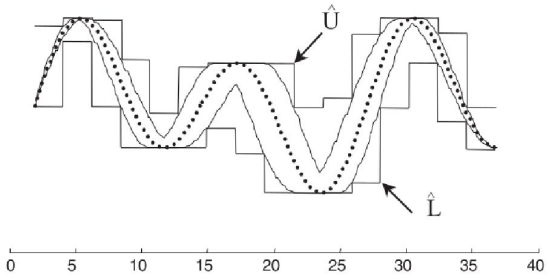

Figura 3.21: Preparación para la indexación de PAA.

Chu [23] en 2002 sugiere un método llamado Iterative Deepening Dynamic Time Warping (IDDTW). Esta técnica utiliza la primera fase de reducción de dimensión del problema que se ha visto en el algoritmo anterior de Keogh. La representación PAA se puede comprimir utilizando cualquier tipo de compresión, y el uso del algoritmo PDTW (Figura 3.22) para obtener una aproximación a la verdadera distancia de DTW.

En 2004, Ratanamahatana [80] propone otra forma de cribar mediante restricciones el espacio de búsqueda. El sistema va aprendiendo y reduciendo las zonas que van a contener la solución óptima. El sistema que propone se ajusta mediante restricciones parametrizadas, con lo que se puede ir ajustando el espacio de búsqueda. En la Figura 3.23a se muestra un ejemplo de cómo el espacio de búsqueda se va modificando.

La Figura 3.24 muestra una breve descripción del funcionamiento de este método. En primer lugar se observan las funciones $h(1)$ y $h(2)$, para compararlas posteriormente. En función del resultado de la comparación realiza una composición de una forma u otra.

Salvador y Chan [87] en 2004 proponen FastDTW que es un algoritmo multinivel con tres niveles:

1. Reducción: Convertir una serie de tiempo en otra más pequeña que representa los mismos datos con la mayor precisión posible y con el menor tamaño.

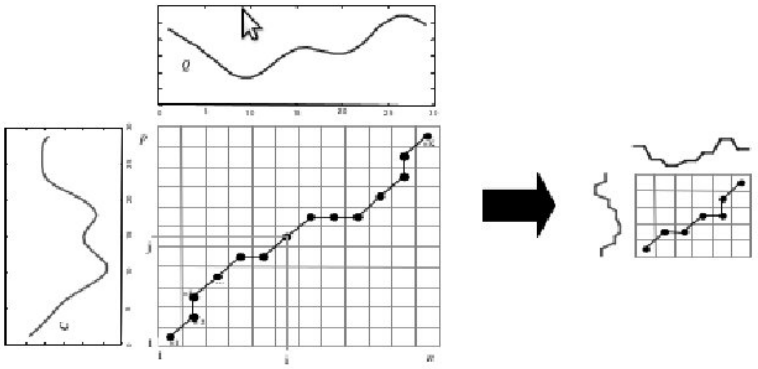

Figura 3.22: A) Ejemplo con DTW. B) PDTW.

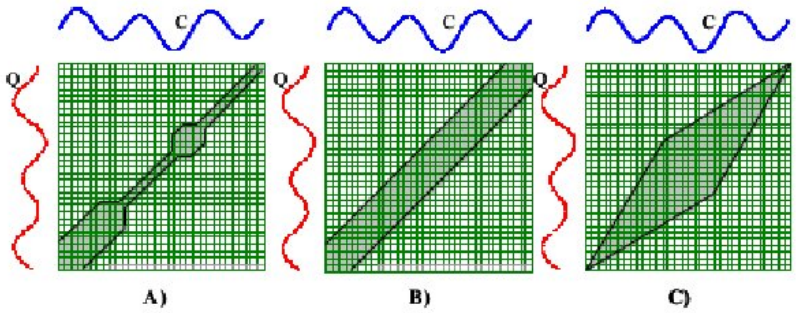

Figura 3.23: A) incremento/decremento ancho de banda. B) Sakoe-Chiba. C) Paralelograma de Itakura.

2. Proyección: Encontrar un camino deformación de distancia mínima en la resolución menor, y utilizar ese camino, como una estimación inicial para calcular la ruta en una resolución más alta.
3. Refinamiento: Define la estimación inicial proyectada a partir de una resolución más baja a través de ajustes locales.

La técnica de reducción se hace combinando una celda con sus adyacentes. Este proceso se repite varias veces, obteniendo diferentes representaciones de los datos para distintas resoluciones. Un punto de la resolución más baja puede equivaler a cuatro puntos de los datos originales. Aunque esta técnica no garantiza se obtenga el camino óptimo, sí que garantiza una solución bastante próxima a la óptima.

En 2005 Shou et al. [89] propone una nueva forma de calcular los límites para DTW basada en la utilización de APCA. En primer lugar, se describe una técnica con la que se aproxima cada secuencia a una secuencia de segmentos M. Este método toma cada grupo de valores y crea una terna compuesta por los valores mínimo, máximo y el número de valores en segmento. En la Figura 3.26 se muestra un ejemplo: la terna tiene los valores

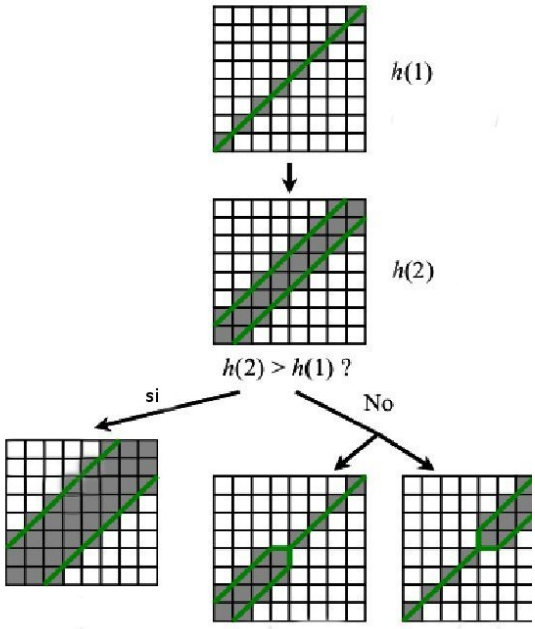

Figura 3.24: Algoritmo del método de Ratanamahatana

(4,8,6), donde 4 es el mínimo, 8 es el máximo y 6 es el número de puntos.

A continuación, se modifica el algoritmo DTW creando el denominado SDTW (Segmented Dynamic Time Warping). Para procesar $Lbseg(\vec{q},\ \vec{s})$ se construye una matriz de NxM, donde el elemento (i, j) contiene la distancia entre los segmentos q_i and s_j, la función de comparación viene definida por la Ecuación 3.30.

$$D(\vec{x}, \vec{y}) \equiv \sqrt{\sum_{t=0}^{n-1} x_t - y_t} \tag{3.30}$$

Para proporcionar un límite inferior que se calcule de forma eficiente utiliza una vesión a del algoritmo DTW. También describe cómo el límite puede ser más reducido en presencia de restricciones de deformación. Finalmente, se desarrolla un índice y una técnica de múltiples pasos que utiliza los límites propuestos y realiza dos niveles de filtrado para procesar de manera eficiente las consultas de similitud.

Sakurai [86] propone un método llamado FTW (Fast Search Method for Dynamic Time Warping) en este mismo año. Esta técnica utiliza un sistema de representación llamado aproximación de segmentos. La Figura 3.27 muestra un ejemplo, cada uno representado por un rango y un intervalo.

La comparación se realiza mediante refinamientos sucesivos. La Figura 3.28 muestra cómo se realizan estos refinamientos. En la parte superior se ve la representación de un dato y una consulta a diferentes escalas. Mientras que en la parte inferior el resultado de

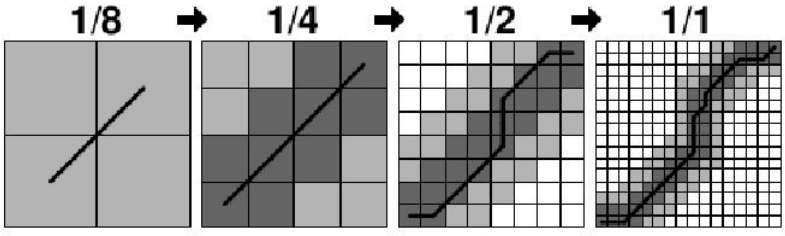

Figura 3.25: Las cuatro diferentes resoluciones durante la evaluación del algoritmo Fast-Dtw.

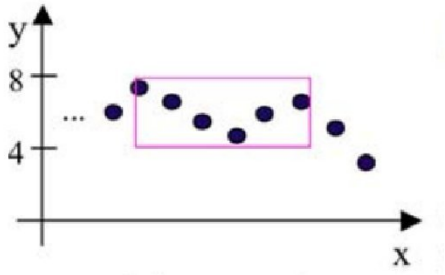

Figura 3.26: Creación de terna.

marcar todas las celdas en las hay alguna parte del dato P^A o de la consulta Q^A.

Para calcular la distancia entre dos puntos se utiliza un algoritmo basado en la distancia a los k-vecinos más próximos. Esta versión del algoritmo permite usar restricciones globales, por lo que se agiliza el cálculo, y se puede combinar con el uso de la matriz de proximidad. La Figura 3.29 muestra un ejemplo de la matriz.

Para realizar el refinamiento proponen el uso de un algoritmo de granularidad simple, aunque se puede usar para granularidad múltiple si se desea.

En este año, Ratanamahatana [81] habla sobre tres mitos que no cumple la técnica DTW. Estos mitos son:

1. La gran ventaja de trabajar con DTW es cuando las secuencias son de longitudes diferentes.
2. La limitación de los caminos de deformación son un mal necesario, a la hora de buscar mejoras en los algoritmos no se deben hacer con restricciones.
3. Hay una necesidad de acelerar el algoritmo para que su velocidad de ejecución sea mayor. Todas las mejoras que se van planteando es en la linea de bajar la complejidad de ejecución a $O(n)$, pero si se utilizan menos delimitaciones se puede ver como la técnica tiene esencialmente $O(n)$.

Para cada un estos mitos demuestra su falsedad con experimento [81].

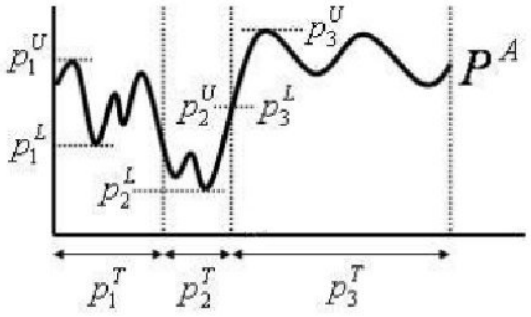

Figura 3.27: Aproximación para agilizar el cálculo por Sakurai.

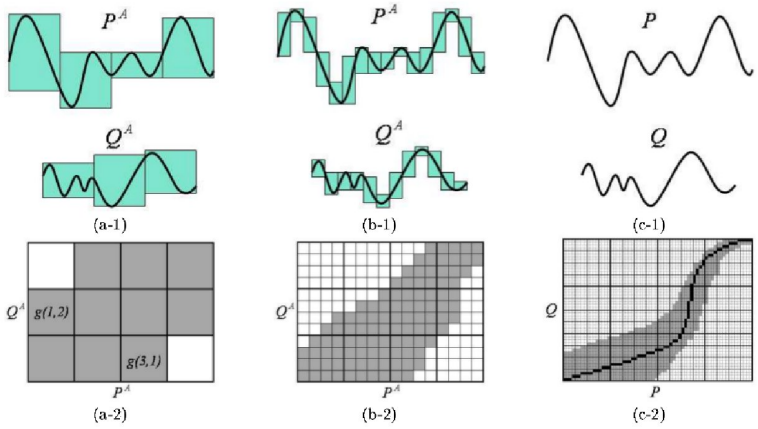

Figura 3.28: Comparación de datos en Sakurai

La estructura secuencial indexada (SIS) es una propuesta de Ruengronghirunya [84] en 2009. El objetivo es buscar el equilibrio entre el costo de E/S y la eficiencia de indexación en la medida de similitud DTW.

3.2.3. Deformación por regresión (RTW - Regression Time Warping).

Lei y Govindaraju [63] en 2004 proponen *Regression Time Warping* (RTW). Este algoritmo consigue ser más rápido que DTW, y más preciso que la distancia euclídea. En determinadas condiciones, su precisión es comparable con DTW. En lo que a velocidad y precisión se refiere consiguen estar entre DTW y la distancia euclídea.

Este método para calcular el camino de deformación utiliza una estrategia local (no global como DTW). La Figura 3.30 muestra cómo sólo se eligen de las celdas adyacentes que van hacia adelante o hacia arriba. Esa estrategia combinada con el uso de restriciones

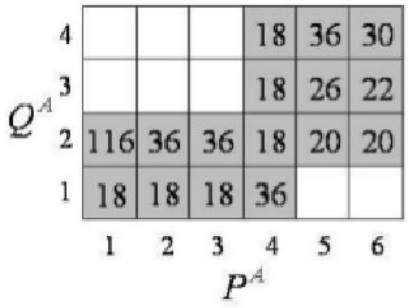

Figura 3.29: Ejemplo de matriz de deformación calculada con segmentos aproximados.

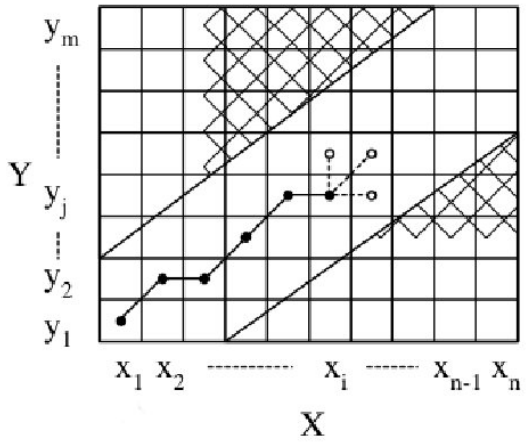

Figura 3.30: Calculo camino.

globales (banda Sakoe-Chiba, paralelogramo Itakura, ...) hace que este método sea mucho más rápido. El ultimo paso es el cálculo de la distancia (Ecuación 3.31).

$$RTW(X,Y) = \frac{1}{k} \sum_{1 \leq i \leq nn, 1 \leq j \leq m} cost(x_i - y_j) \tag{3.31}$$

donde (x_i,y_j) está dentro de las celdas que cumplen las restricciones.

La principal característica de RTW es ser invariante a la escala y el desplazamiento. Aunque es tan frágil como la distancia euclídea, por que sólo permite una coincidencia, obtiene la distancia óptima sin llegar a ordenes de complejidad de n^2.

3.2.4. Longest Common SubSequence (LCSS).

En 2002, Vlachos [98] creó una nueva alternativa a DTW con el objetivo de almacenar datos en dos y tres dimensiones denominada *Función de similitud basada en la subsecuencia común más larga* (LCSS - Longest Common SubSequence). La función de similitud se define con varias opciones:

1. La Ecuación 3.32 que se apoya en la Ecuación 3.33 para definir el encaje de las mismas en caso de estrechamiento.
2. La Ecuación 3.34 permite las posibles traslaciones y se combina con la anterior formado la Ecuación 3.35.
3. Mediante las Ecuaciones 3.36 y 3.37 se controlan los elementos simétricos.

$$S1(\delta, \varepsilon, A, B) = \frac{LCSS_{\delta,\varepsilon}(A, B)}{MIN(N, M)} \tag{3.32}$$

$$k_i = \begin{cases} 0 & \text{si } A \text{ o } B \text{ están vacias,} \\ 1 + LCSS_{\delta,\varepsilon}(head(A), head(B)) & \text{si } |a_{x,n} - b_{x,m}| < \varepsilon \ y \ |n - m| \leq \delta, \\ max(LCSS_{\delta,\varepsilon}(head(A), B)), LCSS_{\delta,\varepsilon}(head(B), A))) & \text{en caso contrario,} \end{cases} \tag{3.33}$$

donde δ indica el máximo tiempo para buscar el punto a coincidir con la trayectoria y ε controla el umbral de adaptación.

$$f_{c,d}(A) = ((a_{x,1} + c, a_{y,1} + d), (a_{x,2} + c, a_{y,2} + d), ..., (a_{x,n} + c, a_{y,n} + d)) \tag{3.34}$$

$$S2_(\delta, \varepsilon, A, B) = max_{f_{c,d}} S1(\delta, \varepsilon, A, f_{c,d}(B)) \tag{3.35}$$

$$D1_(\delta, \varepsilon, A, B) = 1 - S1(\delta, \varepsilon, A, B) \tag{3.36}$$

$$D2_(\delta, \varepsilon, A, B) = 1 - S2(\delta, \varepsilon, A, B) \tag{3.37}$$

3.2.5. Grafo Acíclico Dirigido (DAG - Directed Acyclic Graph).

En 2005, Latecki [61] crea una técnica basada en grafos llamada *grafo acíclico dirigido DAG* (Directed Acyclic Graph). DAG propone un algoritmo de evaluación llamado *Enlace con mínima varianza* (MVM-Minimal Variance Maching) que realiza de forma automática las siguiente tareas:

1. Determina la secuencia que mejor enlaza.
2. Salta automáticamente los valores extremos.
3. Calcula la traducción o la escala de valores correspondientes que minimiza la varianza de las diferencias de los elementos correspondientes.

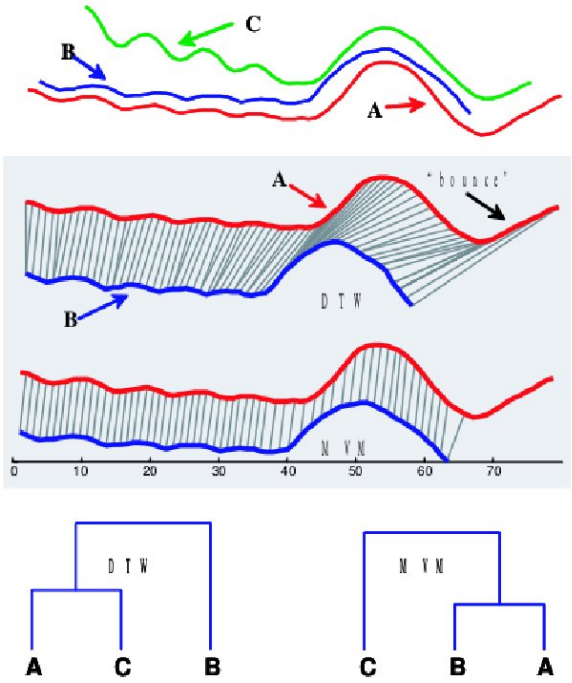

Figura 3.31: Ejemplo DAG.

En el ejemplo de la Figura 3.31 se muestra cómo en la parte derecha de la gráfica sobran algunos valores. Si se utiliza DTW utiliza estos valores y eso hace que se cometan errores en los cálculos. Si se usa la técnica MVM esos valores sobrantes son ignorados por lo que los cálculos son más precisos [61].

3.2.6. Edición en la secuencia real (EDR - Edit Distance on Real sequence).

La técnicas anteriores tienen problemas con el ruido, errores de lectura de los sensores, etc. Chen [18] en 2005 presenta un método llamado *edición en la secuencia real* (EDR - Edit Distance on Real sequence). Esta técnica trata de dar robustez a la formula de la distancia respecto a los errores y problemas de los datos. El resultado obtenido es un sistema que es el 50% más preciso que LCSS e igual de preciso que DTW. Además, también incorporan tres técnicas de poda: valor medio del Q-Gram, triángulo desigual cercano y histogramas [18], haciendo que el algoritmo EDR sea más eficiente. La Ecuación 3.38 es la utilizada por EDR.

$$EDR(R,S) = \begin{cases} n & \text{si m = 0} \\ m & \text{si n = 0} \\ min(P1,P2,P3) & \text{en otro caso} \end{cases} \tag{3.38}$$

donde $P1 = EDR(Resto(R), Resto(S)) + subcoste$, $P2 = EDR(Resto(R), S) + 1$ y $P3 = EDR(R, Resto(S)) + 1$ y $subcoste$ se calcula con la Ecuación 3.39.

$$EDR(R,S) = \begin{cases} 0 & \text{si } match(r_1, s_1) = cierto \\ 1 & \text{en otro caso} \end{cases} \tag{3.39}$$

Si se compara con DTW, LCSS y Distancia Euclídea se le podría otorgar las siguientes bondades:

1. El umbral reduce los efectos de ruido al cuantificar la distancia entre un par de elementos a dos valores, 0 y 1 (LCSS también realiza la misma cuantificación). Por lo tanto, el efecto de los valores extremos en la distancia medida es mucho menor en la EDR que en la distancia euclídea, DTW, y ERP.
2. Como ERP busca el mínimo número de operaciones de edición cuando se va a cambiar de trayectoria.
3. EDR asigna penalizaciones a las ramas cuyos valores sean altos. Esto hace que sea más preciso que LCSS.

3.2.7. Evaluación rápida de series temporales (FTSE - Fast Time Series Evaluation).

Morse y Patel [70] en 2007 proponen una técnica llamada *evaluación rápida de series de tiempo* (FTSE - Fast Time Series Evaluation). Con este método se puede evaluar el valor umbral de distintas técnicas basadas en esta idea de los cuales EDR y LCSS son sólo dos ejemplos. Extiende las técnicas umbrales con un nuevo marco llamado Swale. Los pasos que sigue esta técnica son los siguientes:

- Identificación de elementos que enlazan, para lo que compara cada R_i con cada S_i obteniendo una lista de pares con las mejores asociaciones.
- Establece la puntuación del marco que se está evaluando (LCSS, EDR) o Swale que consiste en crear un array de emparejamiento de longitud n.

Un requisito importante debe ser que la estrategia de indexación para ganar al paradigma de programación dinámica es que el número de celdas del array sea menor que $m * n$, donde m es la longitud de la serie a evaluar y n es la longitud de la consulta. En [70] se pueden ver todos los detalles sobre cómo se puede utilizar FTSE con técnicas como LCSS y EDT.

3.2.8. Segmento de distorsión de tiempo (STW - Segment-wise Time Warping).

Zhao y Wong [106] proponen una solución para el problema de escalado denominada *Segmento de distorsión de tiempo* (STW - Segment-wise Time Warping).

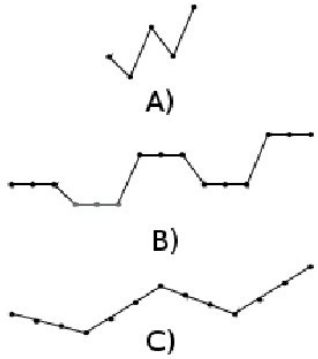

Figura 3.32: Ajuste de datos: (A) Datos originales. (B) Usando PTW. (C) Usando STW.

Como medida de similitud utiliza el cuadrado de la distancia euclídea (Ecuación 3.40) de los dos segmentos. Para este cálculo necesita dos series que tienen el mismo intervalo de tiempo. Para completar el número de datos (en caso de ser necesario) añade los nuevos datos realizando la interpolación de los datos conocidos. La Figura 3.32c muestra un ejemplo de transformación con STW, mientras que la Figura 3.32b se ve como sería con el criterio PTW (no mantiene la forma).

$$d(A,B) = (a_i - b_j)^2 + (a_{i+1} - b_{j+1})^2 \tag{3.40}$$

donde, A(a_i, a_{i+1}) y B(b_i, b_{i+1}) son los segmentos a comparar.

Cuando un segmento ha sido estirado se utiliza la Ecuación de similitud 3.41, que se usa para ver la distancia entre dos segmentos.

$$d(A, B(j \sim j+N)) = \sum_{k=0}^{N} (a_i + \frac{k}{N}(a_{i+1} - a_i) - b_{j+k})^2 \tag{3.41}$$

donde $S(s_1, s_2, ..., s_n)$ y $Q(q_1, q_2, ..., q_n)$ son los segmentos a comparar.

3.2.9. DTW con escala uniforme (SWM - Scaled and Warped Matching).

La Figura 3.33 ilustra un ejemplo sobre cómo se han asociado los puntos de dos gráficas. Una vez realizada esta asociación se define la matriz de distancias con la que se calcula el camino mínimo (Figura 3.34) usando esta ecuación.

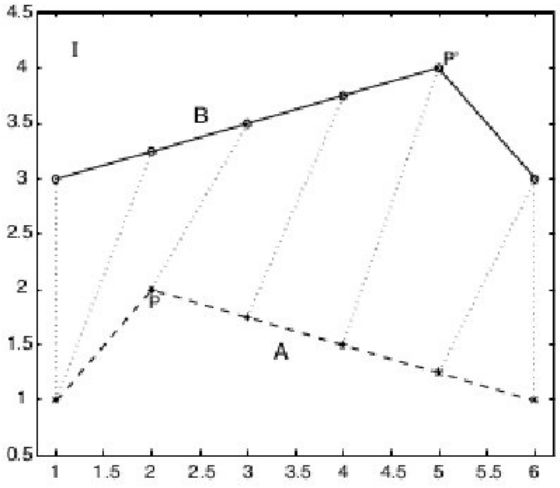

Figura 3.33: Asociación de puntos usando algoritmo STW.

Fu [34] en 2008 crea una técnica llamada *DTW con escala uniforme* (SWM - Scaled and Warped Matching). En este trabajo muestra la importancia del problema de escalado y la distancia de distorsión en las consultas. Dado que el cálculo de la distancia SWM es muy costoso propone una técnica que límita el espacio de búsqueda.

En primer lugar se va a ver el funcionamiento de SWM. Se parte de dos secuencias: una consulta $Q = q_1, q_2, ..., q_n$ y una secuencia a evaluar $C = c_1, c_2, ..., c_n$, como valor a utilizar por el proceso es el factor de escalado que se considera 1, y como restricciones se usa Sakoe-Chiba. Sobre estos datos crea dos secuencias (U, L) y con estos límites poda la secuencia eliminando los puntos que no están dentro de los márgenes permitidos. Se puede llegar a podar más del 90 % del espacio de búsqueda para la búsqueda del vecino más cercano de una en una gran variedad de datos. El método se puede ampliar fácilmente para cubrir la búsqueda de los k vecinos más cercanos. En la Figura 3.35 se puede ver como cambia el realizar la asociación de puntos con el criterio DTW y con el criterio SWM.

3.2.10. Comparación de patrones

Uno de los métodos más utilizados fue propuesto por Berndt y Clifford [9]. Este método se basa en la comparación de patrones, por un lado, se crean una serie de patrones, y por otro, se convierten las secuencias de búsqueda. Para realizar la conversión se deben realizar varios pasos:

1. Comparar los patrones: se obtiene una tabla con los coeficientes de comparación. La Figura 3.36 muestra los cuatro patrones (mnt5, mnt10, mnt20, flat40). La Tabla 3.1 muestra sobre el ejemplo cómo queda la comparación emtre las distintas plantillas.
2. Elegir un coeficiente que fija la plantilla y la serie a utilizar. En el ejemplo se ha elegido 0.85, que selecciona la plantilla **mnt10** y la serie **mnt20**.
3. Finalmente crear la matriz de distancias acumuladas (Tabla 3.2). Sobre esta matriz se

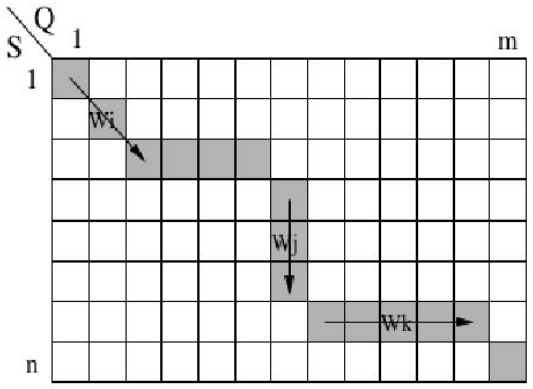

Figura 3.34: Matriz de distancias del algoritmo STW.

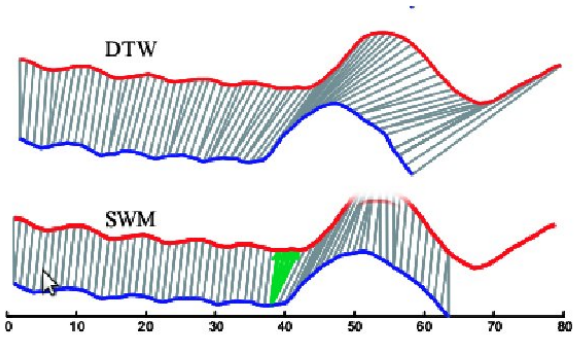

Figura 3.35: Comparación de algoritmos DTW y SWM.

extrae la ruta de deformación, que consiste en una secuencia de pares (i, j) tal que la deformación es mínima. Esta ruta está sometida a las condiciones de contorno, continuidad y monotonía. Además de estas condiciones debe comenzar y acabar en diagonal.

Ruspini y Zwir [96] proponen una forma de procesar las características más importantes de elementos complejos. Los objetivos que persigue esta técnica son dos:

- Calidad de Ajuste.- Mide el parecido de los datos representados a los datos reales.
- Ampliación.- Mide a través de una función lineal la longitud de intervalo que está siendo utilizado.

Utiliza conceptos de lógica difusa para almacenar las características, aunque como método de almacenamiento utiliza PLR (Sección 3.1.6). El uso de la lógica difusa permite que los requesitos se puedan describir de una forma más clara. Además los resultados son más legibles y las funciones utilizadas son más simples.

Tabla 3.1: Coeficientes de comparación de Berndt y Clifford.

plantilla / series	flat40	mnt5	mnt10	mnt20
flat40	1.00	0.86	0.76	0.61
mnt5	0.84	1.00	0.91	0.73
mnt10	0.68	0.89	1.00	**0.85**
mnt20	0.36	0.62	0.81	1.00

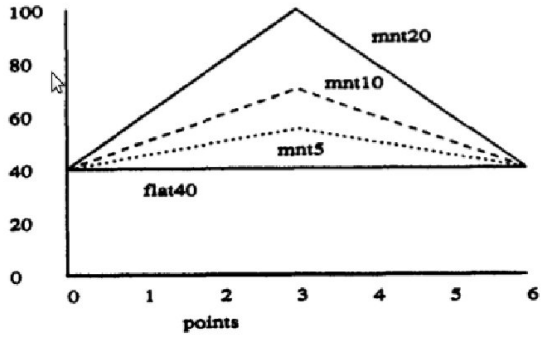

Figura 3.36: Pasos conversión del sistema Berndt y Clifford.

Posteriormente en 2000, Ge y Smyth [37] proponen una técnica basada en módelos de Markov. La idea de este método es descomponer los objetos en componentes individuales y relaciones temporales entre ellos. El algoritmo primero construye el modelo de segmento, seguidamente ejecuta el algoritmo de búsqueda y finalmente normaliza el resultado.

En ese mismo año, Wu et al [100] crea un modelo de recuperación llamado FALCON. El sistema propuesto está diseñado para realizar consultas dentro de espacios métricos. La función de distancia (Ecuación 3.42) depende sólo de la distancia entre los datos, no de la naturaleza de los mismos. El sistema permite etiquetar posibles resultados como buenos, malos, etc. Este sistema permite realizar búsqueda de cualquier tipo de información.

$$(D_G(x))^\alpha = \begin{cases} 0 & \text{si } (\alpha < 0) \wedge \exists i \; d(x, g_i) = 0 \\ \frac{1}{k} X \sum_{i=1}^{k} d(x, g_i)^\alpha & \text{en otro caso.} \end{cases} \tag{3.42}$$

donde G($g_1, g_2, ..., g_n$) es el conjunto de buenas consultas, d es la distancia entre dos objetos y x es el objeto candidato.

También ofrece la posibilidad de funcionar sobre una ventana ω, la modificación que habría que realizar a la ecuación anterior sería mínima, quedaría como muestra la Ecuación 3.43. En [100] se pueden ver más detalles de este sistema.

Tabla 3.2: Matriz de distancias acumuladas de Berndt y Clifford.

6	90	50	70	110	130	90	**70**
5	90	30	50	90	90	**70**	70
4	80	20	40	60	70	**60**	80
3	60	20	**20**	**50**	**60**	70	100
2	30	**10**	30	70	90	90	110
1	**10**	10	40	90	130	130	140
0	0	20	60	120	180	180	180
mnt10/mnt20	0	1	2	3	4	5	6

$$(D_G(x))^\alpha = \frac{1}{\sum_{i=1}^{k} \omega_i} X \sum_{i=1}^{k} \omega_i (d(x, g_i))^\alpha \tag{3.43}$$

donde G($g_1, g_2, ..., g_n$) es el conjunto de buenas consultas, d es la distancia entre dos objetos y x es el objeto candidato.

4

Propuesta de Investigación

En este apartado se presenta la investigación realizada en el Trabajo Fin de Máster. En primer lugar se modelan las dos nuevas formas de representación de la serie así como el modo de obtenerlas (Secciones 4.1 y 4.2). Finalmente se expone la primera aproximación a las consultas realizadas sobre la representación de la serie como Conjunto Ordenado de Segmentos (Sección 4.3).

4.1. Representación como Conjunto Ordenado de Segmentos.

Esta sección muestra nuestra representación de series de tiempo mediante Conjuntos Ordenados de Segmentos así como el método para obtenerla. El método utilizado detecta automáticamente los segmentos que componen la serie obteniendo los intervalos temporales del mismo que se representa mediante una línea recta y su validez temporal. La recta se obtiene utilizando regresión lineal. La representación utilizada es similar a algunas representaciones vistas en el Capítulo 3. El método propuesto es totalmente automático y depende solamente de dos parámetros.

El método utiliza como entrada una serie de tiempo D (Ecuación 4.1).

$$D = \{d_1, d_2, \ldots, d_n\} \tag{4.1}$$

donde d_i es el valor de entrada de la serie en el instante i donde $1 \leq i \leq n$.

La salida es un Conjunto Ordenado de Segmentos S. Cada segmento se puede representar mediante la ecuación de la recta (Ecuación 4.2) con validez en un intervalo de tiempo.

$$y = mx + c \tag{4.2}$$

donde m y c son la pendiente y la constante de la recta respectivamente.

La pendiente de la recta entre dos puntos $A(a_x, a_y)$ y $B(b_x, b_y)$ se calcula mediante la Ecuación 4.3.

$$m = \frac{b_y - a_y}{b_x - a_x} \tag{4.3}$$

Luego un segmento que va desde f hasta l se representará formalmente mediante la Ecuación 4.4.

$$s_{f,l} = \{m_{f,l}, c_{f,l}\} \tag{4.4}$$

donde f y l son los límites del intervalo válido de tiempo del segmento.

El conjunto ordenado de segmentos S se representa con la Ecuación 4.5.

$$S = \{s_{f_1,l_1}, s_{f_2,l_2}, \ldots, s_{f_m,l_m}\} \tag{4.5}$$

donde s_{f_i,l_i} es el segmento i de S.

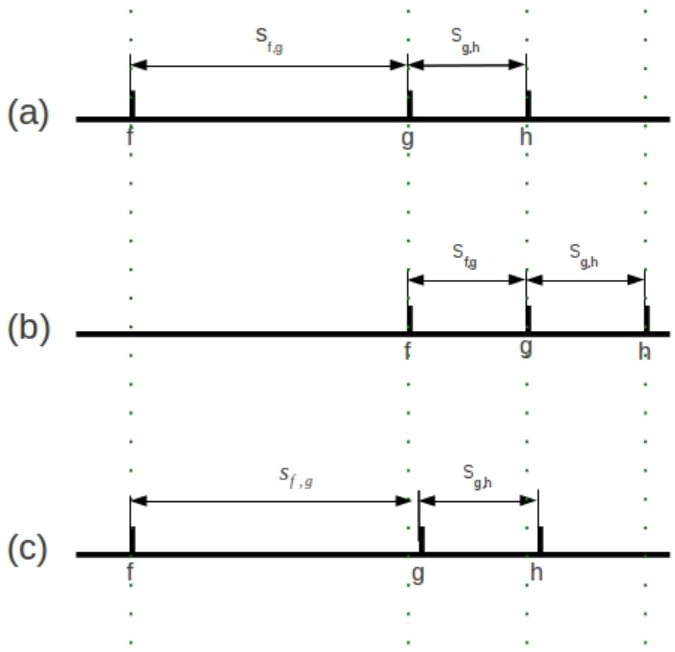

Figura 4.1: (a) El mecanismo de ventana utilizado en nuestra propuesta. (b) Modificación de la ventana cuando $s_{f,g}$ se añade a S. (c) Modificación de la ventana cuando $s_{f,g}$ no se añade a S.

Para calcular S se utiliza en un mecanismo de ventana de tiempo. Es un mecanismo de ventana doble porque se consideran dos tramos (segmentos) en la misma (Figura 4.1a). El parámetro t_w define la longitud mínima de la ventana, es decir, la longitud mínima de los segmentos que componen S. La ventana está definida por tres puntos de tiempo (f, g y h) que definen el intervalo de tiempo de los segmentos $s_{f,g}$ y $s_{g,h}$. El segmento $s_{f,g}$ se

Algoritmo 3 Método Propuesto

1. Definir la ventana de tiempo inicial (Figura 4.1a).
while $h < n$ **do**
  2. Calcular $s_{f,g}$.
  3. Calcular $s_{g,h}$.
  if $|arctan(m_{f,g}) - arctan(m_{g,h})| > \epsilon_m$ **then**
    4. Añadir $s_{f,g}$ a S.
    5. Modificar la ventana de tiempo (Figura 4.1b).
  else
    6. Modificar la ventana de tiempo (Figura 4.1c).
  end if
end while

sitúa temporalmente justo antes del segmento $s_{g,h}$. El método es iterativo, y para marcar la iteración actual se utiliza el índice h.

El Algoritmo 3 muestra el comportamiento del método. El primer paso consiste en definir la ventana de tiempo inicial mediante los instantes f, g y h (Sentencia 1). Esta ventana se define utilizando el parámetro t_w: f, g y h se asignan a 0, t_w-1 y $(t_w-1)*2$ respectivamente. El primer segmento comienza en el primer instante (instante 0) y tiene la mínima longitud t_w, luego g se asigna a $t_w - 1$. El segundo segmento comienza donde acaba el anterior, también tiene la longitud mínima t_w, luego h se asigna a $(t_w - 1) * 2$.

El bucle toma la entrada h (d_h) y la procesa. Primeramente se calculan los segmentos $s_{f,g}$ y $s_{g,h}$ (Sentencias 2 y 3). Se utiliza regresión lineal [88] para representar una nube de puntos, instantes temporales en nuestro caso, mediante una recta. Formalmente, para representar el segmento $s_{f,g}$ se emplea la recta $L_{f,g} = (m_{f,g} * x) + c_{f,g}$ que se calcula utilizando las entradas desde d_f a d_g, luego $s_{f,g} = \{m_{f,g}, c_{f,g}\}$ (Sentencia 2). El mismo proceso se realiza para calcular $s_{g,h}$ (Sentencia 3).

Para determinar si el segmento $s_{f,g}$ debe añadirse a S se utiliza el umbral de tolerancia ϵ_m que define la distancia máxima permitida entre los ángulos de inclinación de los segmentos $s_{f,g}$ y $s_{g,h}$ (Sentencia if). Si la diferencia entre los ángulos de inclinación de los segmentos $s_{f,g}$ y $s_{g,h}$ es mayor que ϵ_m ($|arctan(m_{f,g}) - arctan(m_{g,h})| > \epsilon_m$) entonces $s_{f,g}$ se añade a S (Sentencia 4) y, la ventana temporal se modifica (Sentencia 5). La Figura 4.1b muestra cómo se realiza el proceso. Dado que el segmento $s_{f,g}$ se ha añadido a S, el nuevo $s_{f,g}$ ocupa la posición del antiguo $s_{g,h}$, y el nuevo $s_{g,h}$ pasa a ocupar la posición siguiente a $s_{f,g}$ con un tamaño t_w, luego:

1. $f = g$.
2. $g = f + t_w - 1$.
3. $h = g + (t_w - 1)$.

En otro caso, se modifica la ventana añadiendo un nuevo instante al segmento $s_{f,g}$ (Sentencia 6). La Figura 4.1c muestra cómo se realiza el proceso. El tamaño de $s_{f,g}$ se incrementa en uno y $s_{g,h}$ se desplaza una posición hacia delante, luego:

1. f mantiene su valor.
2. $g = g + 1$.

Tabla 4.1: Conjunto S de salida.

s_{g_i,f_i}	$m_{f,g}$	$c_{f,g}$
$s_{0,\ 8{,}25}$	$0{,}015$	$-0{,}020$
$s_{8{,}25,\ 12{,}65}$	$0{,}208$	$-1{,}610$
$s_{12{,}65,\ 17{,}48}$	$-0{,}173$	$3{,}202$
$s_{17{,}48,\ 20{,}01}$	$-0{,}047$	$1{,}007$
$s_{20{,}01,\ 22{,}54}$	$0{,}096$	$-1{,}854$
$s_{22{,}54,\ 26{,}00}$	$0{,}221$	$-4{,}672$
$s_{26{,}00,\ 29{,}77}$	$-0{,}221$	$6{,}820$
$s_{29{,}77,\ 37{,}00}$	$-0{,}041$	$1{,}471$

3. $h = h + 1$.

Este proceso se realiza mientras hay entradas en la serie.

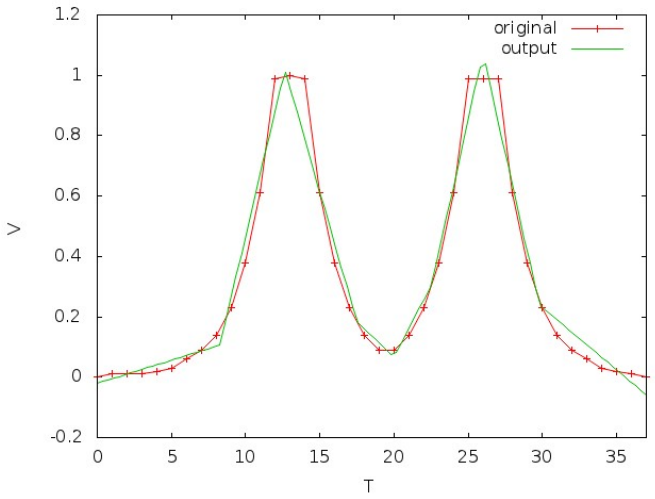

Figura 4.2: Gráfica Serie de Tiempo/Conjunto S obtenido.

Para terminar se mostrará un ejemplo. Sea la serie de tiempo D de entrada que se muestra rotulada como *original* en la Figura 4.2, al aplicar el método presentado se obtiene la salida que muestra rotulada como *output* en la Figura 4.2. La Tabla 4.1 contiene la representación de cada segmento en la notación utilizada en este trabajo (Ecuación 4.4). La primera columna representa la denominación del segmento pudiéndose ver el último instante del mismo calculado por el algoritmo de inducción, la segunda y tercera columnas contienen la pendiente y constante de la recta que representa al segmento respectivamente. Por ejemplo, el segmento $s_{8{,}25,12{,}65}$ representa los valores de la serie desde los instantes del $8{,}25$ al $12{,}65$, y la ecuación que define la recta para ese segmento es $(0{,}208 \times x) + -1{,}610$.

4.2. Representación como Conjunto Ordenado de Estructuras Difusas T.

La nueva representación de la serie hará uso de la lógica difusa para tratar la incertidumbre existente. Esto permitirá comparar el funcionamiento de las consultas sobre ambas representaciones, sin y con lógica difusa. Esta novedosa representación se modelizará mediante un Conjunto Ordenado de Tendencias. Cada tendencia indicará la dirección de los segmentos, y serán de tipo Incremental (INC), Decremental (DEC) o paralelas al eje X ($ZERO$). Esto se puede identificar utilizando las pendientes de los segmentos de S.

La modelización del conjunto de tendencias se hará mediante la tupla $T = \{t_0, LT\}$, donde t_0 es un conjunto difuso que indica el instante temporal donde empieza la lista de tendencias, y $LT =< TEND_1, TEND_2, \ldots, TEND_n >$ es la lista de tendencias propiamente dicha.

Cada elemento de LT se modela mediante la tupla $TEND_i = \{type_i, t_i, v_i, power_i\}$ donde:

- $type_i$ indica el tipo de tendencia tomando uno de los valores $ZERO$, INC o DEC.
- t_i es un número difuso triangular que define el instante temporal donde acaba el intervalo que representa el segmento. El origen del segmento es t_{i-1} de $TEND_{i-1}$.
- v_i es un número difuso que define el valor final de salida del último valor del intervalo. El valor del primer punto del segmento se puede obtener de v_{i-1} de $TEND_{i-1}$.
- $power_i \in SLL$ es una etiqueta lingüística que modela la potencia de incremento o decremento. El conjunto ordenado de etiquetas lingüísticas SLL se define apriori. Se utiliza la pendiente de la recta para calcular esta etiqueta.

Algoritmo 4 Cálculo de $type_i$.

```
if m <0 then
  type_i = DEC
else
  if m >0 then
    type_i = INC
  else
    type_i = ZERO
  end if
end if
```

Para estudiar el cálculo de las tendencias se detallará cómo se obtiene cada componente de T. LT está formado por un conjunto de elementos $TEND_i$ que tienen varios componentes. Cada $TEND_i$ se obtiene de la siguiente forma:

$type_i$: toma uno de los valores $ZERO$, INC o DEC. Este valor está en función de la pendiente de la recta que modela. La pendiente de una recta es mayor o menor que 0 si la recta es creciente o decreciente respectivamente. En caso de pendiente 0 significa que la recta es paralela al eje X (Algoritmo 4).

t_i: Para el cálculo del soporte de este número difuso se utiliza la Ecuación 4.6 que mide el error de la recta obtenida respecto a los puntos originales, y se ha denominado

porcentaje de diferencias. Esta ecuación calcula el porcentaje del error de la recta respecto a cada uno de los puntos originales.

$$p_{f,l} = \frac{\sum_{i=f}^{l} \frac{|r(t_i)-d_i|}{d_i}}{l-f+1} \tag{4.6}$$

donde $r(t_i)$ es el valor del segmento $s_{f,l}$ en el tiempo t_i, d_i es el valor de la serie en el instante t_i, y, f y l son los instantes de comienzo y fin de la recta.

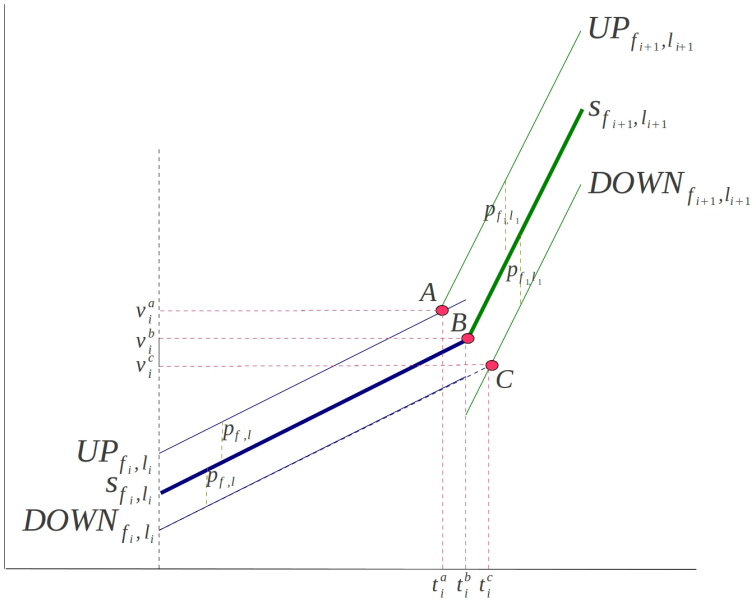

Figura 4.3: Método de cálculo de t_i.

La Figura 4.3 muestra cómo se calculan los puntos que definen el soporte de t_i. Utiliza cuatro rectas, dos rectas paralelas a s_{f_i,l_i} denominadas UP_{f_i,l_i} y $DOWN_{f_i,l_i}$, y dos paralelas a $s_{f_{i+1},l_{i+1}}$ denotadas por $UP_{f_{i+1},l_{i+1}}$ y $DOWN_{f_{i+1},l_{i+1}}$. El valor $p_{f,l}$ se utiliza para calcular las rectas UP_{f_i,l_i} y $UP_{f_{i+1},l_{i+1}}$ desplazadas hacia arriba en $p_{f,l}$ unidades respecto a s_{f_i,l_i} y $s_{f_{i+1},l_{i+1}}$, es decir, $UP_{f_i,l_i} = (m_{f_i,l_i} \times x) + (c_{f_i,l_i} + p_{f,l})$ y $UP_{f_{i+1},l_{i+1}} = (m_{f_{i+1},l_{i+1}} \times x) + (c_{f_{i+1},l_{i+1}} + p_{f,l})$. También se usa para obtener las rectas $DOWN_{f_i,l_i}$ y $DOWN_{f_{i+1},l_{i+1}}$ desplazadas hacia abajo $p_{f,l}$ unidades respecto a s_{f_i,l_i} y $s_{f_{i+1},l_{i+1}}$, es decir, $DOWN_{f_i,l_i} = (m_{f_i,l_i} \times x) - (c_{f_i,l_i} + p_{f,l})$ y $DOWN_{f_{i+1},l_{i+1}} = (m_{f_{i+1},l_{i+1}} \times x) - (c_{f_{i+1},l_{i+1}} + p_{f,l})$.

Sean los puntos $A = \{t_i^a, v_i^a\}$, $B = \{t_i^b, v_i^b\}$ y $C = \{t_i^c, v_i^c\}$. t_i^a se calcula mediante la coordenada X del punto de corte de las rectas UP_{f_i,l_i} y $UP_{f_{i+1},l_{i+1}}$ (Figura 4.3). De igual forma, se obtiene t_i^c mediante la coordenada X del punto de corte de las rectas $DOWN_{f_i,l_i}$ y $DOWN_{f_{i+1},l_{i+1}}$. t_i^b se calcula mediante la coordenada X del punto de corte de las rectas s_{f_i,l_i} y $s_{f_{i+1},l_{i+1}}$. Estos tres puntos definen el soporte del conjunto difuso t_i. Destacar que no tienen porqué estar ordenados. La ordenación se realiza respecto a sus valores numéricos, $t_i = ORDENADOS(t_i^a, t_i^b, t_i^c)$.

v_i: Este conjunto difuso triangular está formado por los valores $\{v_i^a, v_i^b, v_i^c\}$. Cada uno de estos puntos se calcula como el valor que ofrecen las rectas UP_{f_i,l_i}, s_{f_i,l_i} y $DOWN_{f_i,l_i}$ para los valores t_i^a, t_i^b y t_i^c respectivamente, es decir:

Tabla 4.2: Conjunto de etiquetas SLL.

nombre	a	b	c	d
LEVE	0	0	0	10
MEDIO	0	10	20	30
BRUSCO	20	30	INF	INF

Tabla 4.3: Conjunto de Tendencias T obtenido.

$TEND_i$	$type_i$	t_i			v_i			$power_i$
1	INC	6,30	8,25	10,20	0,06	0,11	0,15	LEVE
2	INC	10,61	12,65	14,68	0,21	1,02	1,83	MEDIO
3	DEC	15,94	17,48	19,02	−0,47	0,19	0,84	MEDIO
4	DEC	16,67	20,01	23,35	−0,29	0,07	0,417	LEVE
5	INC	21,64	22,55	23,45	−0,06	0,31	0,682	MEDIO
6	INC	23,78	26,00	28,22	0,19	1,07	1,962	MEDIO
7	DEC	28,99	29,78	30,56	−0,52	0,24	1,00	MEDIO
8	DEC	35,90	37,00	38,10	−0,91	−0,06	0,79	LEVE

1. $v_i^a = UP_{f_i,l_i}(t_i^a)$, valor de la recta UP_{f_i,l_i} para $x = t_i^a$.
2. $v_i^b = s_{f_i,l_i}(t_i^b)$, valor de la recta s_{f_i,l_i} para $x = t_i^b$.
3. $v_i^c = DOWN_{f_i,l_i}(t_i^c)$, valor de la recta $DOWN_{f_i,l_i}$ para $x = t_i^c$.

Al igual que en el caso anterior, los valores pueden estar desordenados y se deben ordenar.

$power_i$: Para calcular el conjunto difuso que define la potencia de la pendiente se utiliza un conjunto de etiquetas lingüísticas definidas apriori, el conjunto $SLL = \{LL_1, \ldots, LL_{n_s}\}$. $power_i$ se asigna a la etiqueta con máximo grado de pertenencia a la pendiente m_{f_i,l_i} de s_{f_i,l_i} (Ecuación 4.7).

$$power_i = argmax_{LL_k} \ \ \mu_{LL_k}(m_{f_i,l_i}) \tag{4.7}$$

Para terminar se mostrará un ejemplo del método presentado. Como entradas necesita un conjunto S y un conjunto de etiquetas lingüísticas SLL, y como salida obtiene un conjunto de tendencias T. Las tablas 4.1 y 4.2 se utilizarán como conjuntos S y SLL de entrada respectivamente. La Tabla 4.3 muestra la salida obtenida. Como puede verse, el tipo se representa mediante los identificadores INC o DEC, la columna t_i representa el conjunto difuso triangular del tiempo donde acaba el segmento, la columna v_i representa el último valor también como conjunto difuso triangular, y $power_i$ toma el valor de una de las etiquetas de SLL.

4.3. Consultas sobre S.

En este apartado se presentará una primera aproximación a las consultas sobre la representación de la serie como un Conjunto Ordenado de Segmentos. También se expondrán

las funciones de comparación utilizadas durante las consultas. Finalmente se mostrará el funcionamiento de estas consultas sobre datos reales.

4.3.1. Formato de los datos.

Se utilizarán tres tipos de elementos. El primero es la lista SP obtenida a partir de S que se usa como entrada al sistema, el segundo es la propia consulta Q, y el tercero es una lista R con los resultados obtenidos. Dado el lenguaje de programación que se ha utilizado para la experimentación, PROLOG, se utilizarán listas de elementos como estructura de datos principal para SP, Q y R.

El conjunto SP quedará representado mediante una lista de elementos $sp_1, sp_2, ..., sp_n$. Cada elemento de esta lista sp_i se representará como muestra la Ecuación 4.8.

$$sp(c_i, d_i, angulo_i) \tag{4.8}$$

donde c_i es la constante de la recta, d_i es la duración del intervalo de tiempo medido en unidades temporales que representa este segmento, y $angulo_i$ es el ángulo respecto a la horizontal que tiene este segmento.

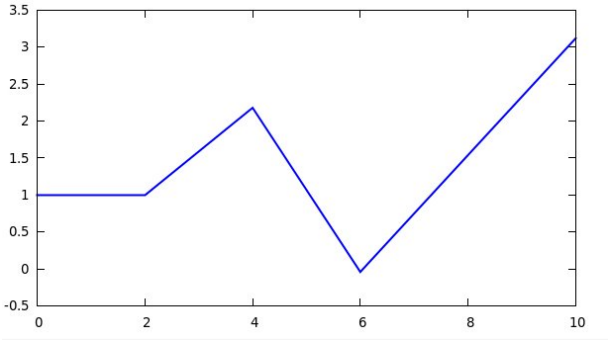

Figura 4.4: Representación de datos.

La serie de datos D se puede representar de dos formas: (1) gráficamente como se muestra en la Figura 4.4; (2) Mediante sintaxis PROLOG (Ecuación 4.9).

$$SP = [sp(1,2,0), sp(1,2,0{,}59), sp(4,2,-1{,}11), sp(3,4,0{,}79), sp(7,2,1{,}25)] \tag{4.9}$$

Para representar la consulta Q se utiliza una representación parecida a la anterior. La estructura de cada hecho de la lista Q será una lista formada por los elementos $q_1, q_2, ...q_n$, donde q_i se representa mediante la Ecuación 4.10.

$$q(c_i, d_i, angulo_i) \tag{4.10}$$

donde c_i es la constante de la recta, d_i es la duración del intervalo de tiempo medido en unidades temporales que representa este segmento, y $angulo_i$ es el ángulo respecto a la horizontal que tiene este segmento.

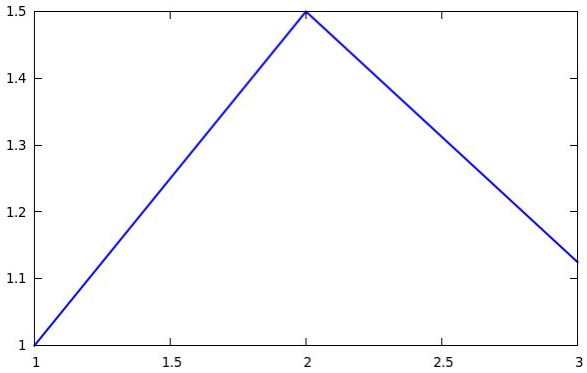

Figura 4.5: Representación de consulta.

La consulta se puede representar también como una gráfica (Figura 4.5) y como un hecho:

$$Q = [q(1, 1, 0.5), q(2, 1.5, 0.25)]$$

La lista R de resultados es una lista de hechos que comienzan por la letra r. La Ecuación 4.11 muestra su estructura.

$$r(p_i, v_i, d_i) \tag{4.11}$$

donde p_i indica en que posición se encuentra esa solución, v_i es la parte de la recta que está comparando y d_i es la distancia encontrada entre los dos tramos.

Aquí vemos el ejemplo de un hecho resultado:

$R = [r(0, [sp(0.0113, -0.0108, 4)], 14.117),$
$r(1, [sp(0.0113, -0.0108, 4)], 14.117),$
$r(2, [sp(0.0113, -0.0108, 4)], 14.117),$
$r(3, [sp(0.0113, -0.0108, 4)], 14.117),$
$r(4, [sp(0.0113, -0.0108, 3), sp(0.096, -0.606, 1)], 7.058), \ldots$

4.3.2. Procedimiento para realizar las consultas.

A continuación se mostrará el modo de realizar las consultas. Primeramente se muestran los bloques que componen el sistema (Figura 4.6):

Extracción de características: Se encarga de tomar los datos y adaptarlos al formato de hechos presentado anteriormente. También toma las listas en el formato de las Ecuaciones 4.8 y 4.10, y transforma sus hechos para que haya un elemento en la lista por cada unidad temporal, lo que permite simplificar el proceso de comparación.

Proceso de comparación: Realiza la comparación entre SP y Q (Figura 4.7). Se debe indicar que tipo de comparación se está realizando.

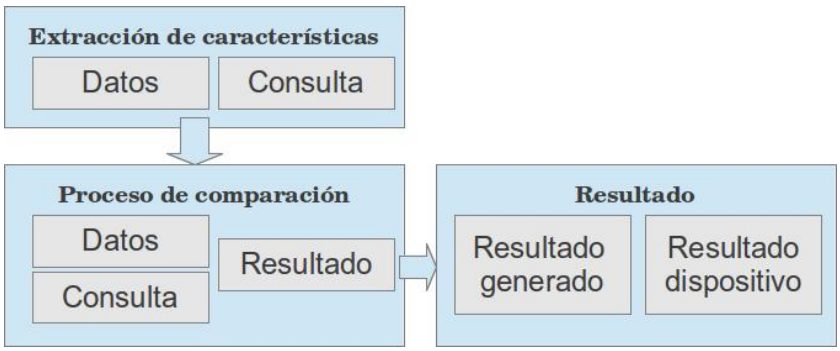

Figura 4.6: Diagrama Bloques del Sistema.

Resultado: Se encarga de organizar y mostrar los resultados unificando las listas de resultados siempre que sea necesario.

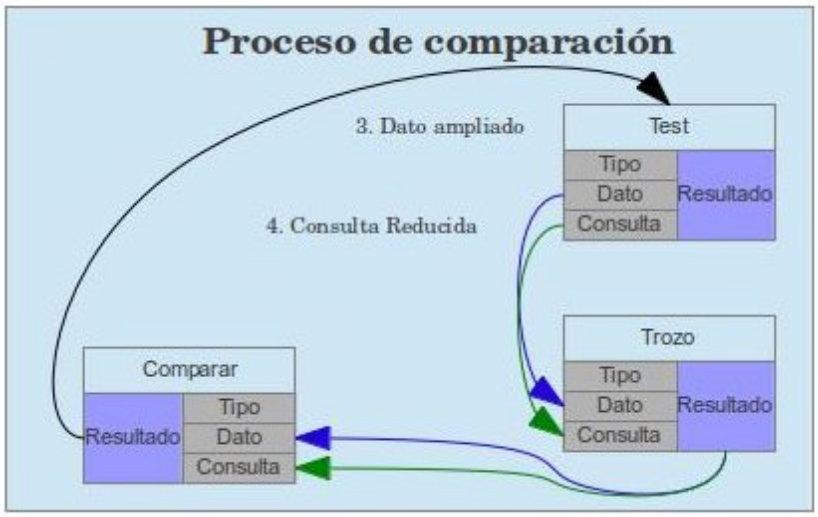

Figura 4.7: Proceso de Comparación.

A continuación se detallará el proceso de comparación que es el de mayor interés (Algoritmo 5). Es un algoritmo recursivo que se denomina *Test*. Este algoritmo realiza un recorrido por SP y va comparando todas las posibles soluciones. La Sentencia 1 extrae una subcadena de SP mediante el Algoritmo Extraer (Algoritmo 6).

La Sentencia 2 debe utilizar un criterio de comparación. Se han estudiado tres:

- Binario.- Retorna cierto si son iguales y falso si son distintos (Ecuación 4.12).

$$f(v_1, v_2) = \left\{ \begin{array}{ll} 0 & \textbf{si } v_1 \neq v_2 \\ 1 & \textbf{si } v_1 = v_2 \end{array} \right\} \tag{4.12}$$

Algoritmo 5 Test.

1. Extraer un trozo de tamaño igual que la consulta (Algoritmo 6).
2. Compara el trozo con la función elegida.

if (Hay más posibles subcadenas) **then**

3. Realiza una llamada a sí mismo con una lista más pequeña.

end if

4. Construye la solución.

Algoritmo 6 Extraer.

if (Longitud es 0) **then**

1. Retorna una lista vacia.

end if

2. Se lLama a sí mismo con Longitud -1, y una lista con un elemento menos.
3. Construye la solución.

- Sin umbral.- Devuelve el valor absoluto de la diferencia de los dos ángulos (Ecuación 4.13).

$$f(v_1, v_2) = |v_1 - v_2| \tag{4.13}$$

- Con umbral.- Retorna *umbral* si la diferencia entre los dos ángulos es mayor que un valor umbral, sino se retorna el valor de la diferencia (Ecuación 4.14).

$$f(v_1, v_2, umbral) = \left\{ \begin{array}{ll} umbral & \text{si } |v_1 - v_2| > umbral \\ |v_1 - v_2| & \text{si } |v_1 - v_2| \leq umbral \end{array} \right\} \tag{4.14}$$

4.3.3. Ejemplo.

Nuestro método necesita dos entradas, una lista SP que se obtiene a partir del Conjunto Ordenado de Segmentos S, y la propia consulta (Sección 4.3.2). La Tabla 4.1 muestra el conjunto S de forma numérica, mientras que la Figura 4.2 muestra la representación gráfica de S. La lista SP de hechos utilizada como entrada al sistema será la siguiente:

$SP = [sp(0.0113, -0.0108, 7.0275), sp(0.0960, -0.6060, 9.5703),$
$sp(0.2240, -1.8310, 13.0000), sp(-0.2240, 3.9930, 16.4297),$
$sp(-0.0960, 1.8900, 18.9930), sp(0.0470, -0.8260, 22.1569),$
$sp(0.2510, -5.3460, 25.4453), sp(-0.1600, 5.1120, 30.8437),$
$sp(-0.0960, 3.1380, 31.8400), sp(-0.0174, 0.6363, 37.0000)]$

Otra entrada al sistema es la consulta a realizar (Figura 4.8), y su representación será:

$Q = [q(0.0113, -0.0108, 2), q(0.0800, -0.6060, 2)]$

Una vez aplicado el método presentado el resultado obtenido es el siguiente:

$R = [r(7, [sp(0.0113, -0.0108, 2), sp(0.0960, -0.6060, 2)], 2.6667),$
$r(6, [sp(0.0113, -0.0108, 3), sp(0.0960, -0.6060, 1)], 7.0583),$
$r(8, [sp(0.0113, -0.0108, 1), sp(0.0960, -0.6060, 3)], 9.725),$
$r(0, [sp(0.0113, -0.0108, 4)], 11.4500), ...$

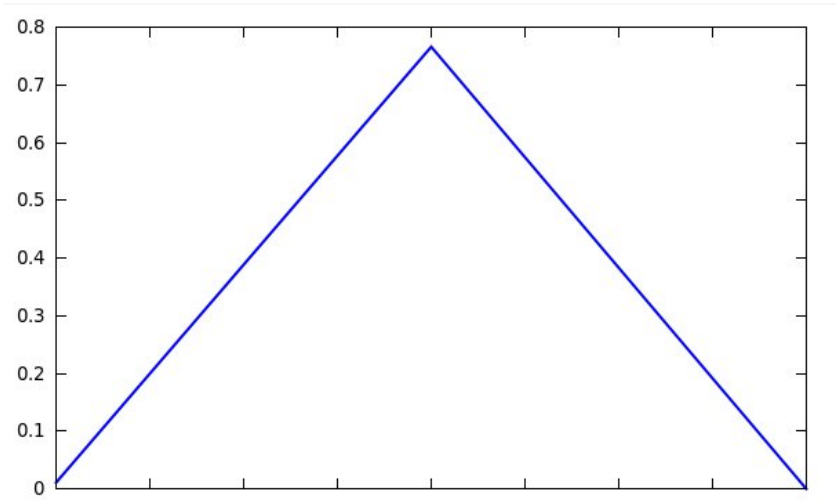

Figura 4.8: Consulta Q del ejemplo.

La Figura 4.9 muestra gráficamente la superposición de cada solución sobre los datos iniciales, donde:

r_1 = $r(7, [sp(0.0113, -0.0108, 2), sp(0.0960, -0.6060, 2)], 2.6667)$.

r_2 = $r(6, [sp(0.0113, -0.0108, 3), sp(0.0960, -0.6060, 1)], 7.0583)$.

r_3 = $r(8, [sp(0.0113, -0.0108, 1), sp(0.0960, -0.6060, 3)], 9.725)$.

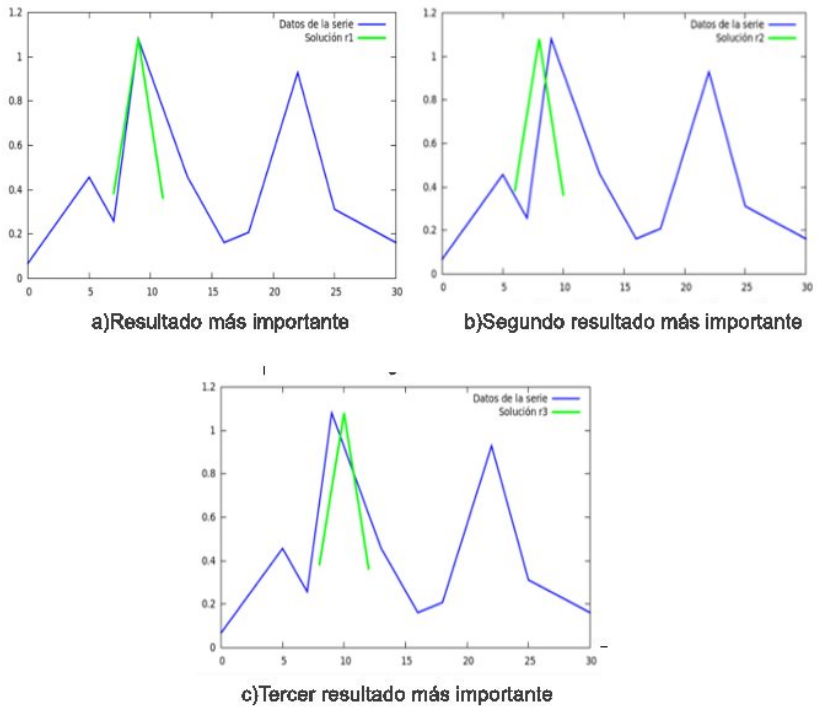

Figura 4.9: Resultado R del ejemplo.

5

Resultados Preliminares

En esta sección se van a presentar los resultados preliminares obtenidos. Se realizarán un conjunto de tres test consistentes en tres pruebas diferentes en cada test. Las series representan la temperatura media mensual del Castillo de Nottingham de 1920 a 1922 (Sección 5.1), número de matrimonios anuales en Escocia desde 1855 a 2011 (Sección 5.2) y número de divorcios anuales desde 1855 a 2011 (Sección 5.3).

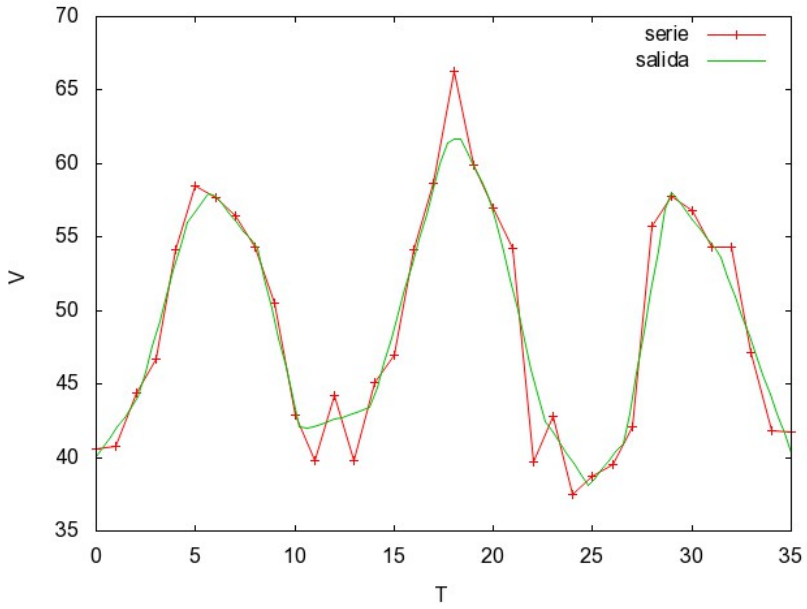

Figura 5.1: Test Temperatura del Castillo de Nottingham con $t_w = 3$ y $\epsilon_m = 5$.

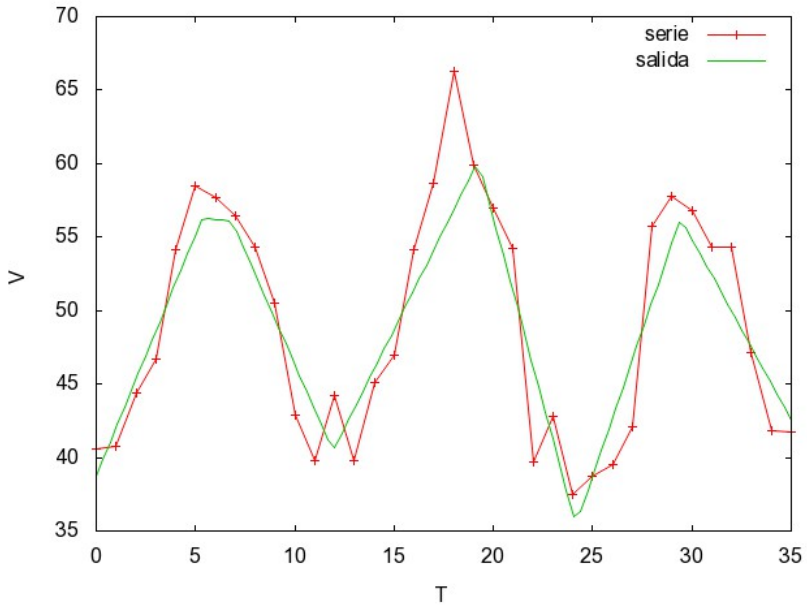

Figura 5.2: Test Temperatura del Castillo de Nottingham con $t_w = 5$ y $\epsilon_m = 5$.

5.1. Temperatura Media Mensual del Castillo de Nottingham de 1920 a 1922.

En esta primera prueba se ha realizado un estudio de la temperatura media mensual del Castillo de Nottingham. La fuente de los datos ha sido *http://datamarket.com/data/set/22li/mean-monthly-air-temperature-deg-f-nottingham-castle-1920-1939*. Tiene información desde el año 1920 a 1939 en grados Farenheit. En la experimentación se ha utilizado la información relativa a los años 1920, 1921 y 1922, es decir, 36 registros. La Figura 5.1 muestra la serie de entrada etiquetada como *serie*. Se han realizado tres pruebas diferentes utilizando una diferencia de ángulos fija ϵ_m asignada a 5 y un tamaño de ventana t_w que ha tomado los valores 3, 5 y 7 para cada una de las pruebas.

Una vez aplicado el Algoritmo 3 sobre la serie de entrada con $t_w = 3$, $t_w = 5$ y $t_w = 7$ respectivamente se obtienen los Conjuntos Ordenados de Segmentos S que muestra en las tres primeras columnas de las Tablas 5.1, 5.2 y 5.3. Las Figuras 5.1, 5.2 y 5.3 muestran la comparativa de la serie de entrada con el resultado obtenido para cada una de las pruebas. Las salidas obtenidas están etiquetadas como *salida*. El error cuadrático medio obtenido (ECM) para cada una de las pruebas es $1{,}9686$, $3{,}2291$ y $4{,}6789$. Finalmente, al aplicar el método de conversión de S a T (Sección 4.2) se obtiene el Conjunto Ordenado de Estructuras Difusas T que muestran las columnas de la cuarta en adelante en las Tablas 5.1, 5.2 y 5.3.

Como conclusiones respecto a la representación como Conjunto Ordenado de Segmentos se puede concluir que:

1. Cuanto menor es el tamaño de la ventana más segmentos se usan para representar la serie.

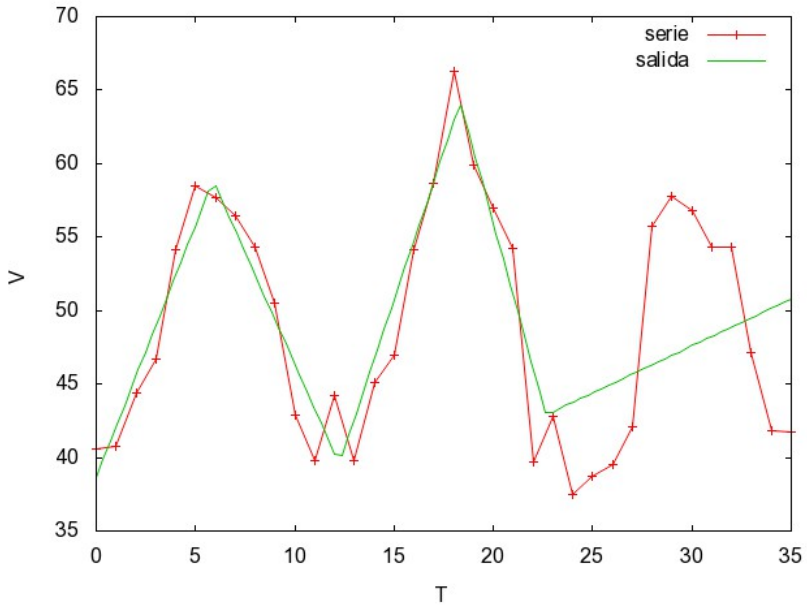

Figura 5.3: Test Temperatura del Castillo de Nottingham con $t_w = 7$ y $\epsilon_m = 5$.

2. Cuanto menor es el tamaño de la ventana menos error se obtiene.
3. En las tres figuras se observa como la representación genera "montañas" para representar cada uno de los años: la subida representa la subida de las temperaturas desde enero hasta el verano, y el descenso la bajada de temperaturas desde el verano hasta fin de año.
4. La Figura 5.3 representa incorrectamente el último año. Esto se debe a que la longitud mínima de la ventana ($t_w = 7$) es bastante grande teniendo en cuenta el número de entradas de la serie, y eso provoca que se utilicen un mayor número de entradas en cada segmento forzando a ignorar máximos y mínimos locales. Este error se debe estudiar en trabajos futuros.
5. Destacar que para $t_w = 3$ y $t_w = 5$ se obtienen buenas representaciones.

Como conclusiones respecto a la representación como Conjunto Ordenado de Estructuras Difusas se puede concluir:

1. Sólo hay tendencias DEC e INC debido a la forma de la gráfica.
2. Los números difusos t_i e v_i obtenidos para todos los casos son coherentes, lo que indica que la metodología presentada en la Sección 4.2 obtiene buenos resultados.
3. La potencia de la pendiente $power_i$ obtenida es en casi todos los casos $BRUSCO$. Esto se debe a la fuerte pendiente de las gráficas, luego es un resultado coherente para estas pruebas.

s_{g_i,f_i}	$m_{f,g}$	$c_{f,g}$	**T**	$type_i$	t_i			v_i			$power_i$
$s_{0,00,\ 2,10}$	1,90	40,03	1	INC	2,07	2,10	2,12	43,34	44,02	44,69	$BRUSCO$
$s_{2,10,\ 4,56}$	4,85	33,85	2	INC	4,51	4,56	4,62	55,43	55,98	56,53	$BRUSCO$
$s_{4,56,\ 5,79}$	1,80	47,77	3	INC	5,58	5,79	6,00	57,61	58,19	58,77	$BRUSCO$
$s_{5,79,\ 8,13}$	−1,70	68,03	4	DEC	7,75	8,13	8,50	53,36	54,22	55,08	$BRUSCO$
$s_{8,13,\ 10,30}$	−5,70	100,53	5	DEC	10,25	10,30	10,35	39,83	41,84	43,85	$BRUSCO$
$s_{10,30,\ 10,17}$	0,65	35,15	6	INC	7,69	10,17	12,65	41,56	41,76	41,96	$BRUSCO$
$s_{10,17,\ 13,87}$	0,45	37,18	7	INC	13,33	13,87	14,41	41,76	43,43	45,09	$MEDIO$
$s_{13,87,\ 17,63}$	4,79	−23,02	8	INC	16,92	17,63	18,33	57,58	61,41	65,24	$BRUSCO$
$s_{17,63,\ 18,32}$	0,60	50,83	9	INC	17,60	18,32	19,04	59,73	61,82	63,92	$BRUSCO$
$s_{18,32,\ 19,98}$	−2,85	114,03	10	DEC	14,58	19,98	25,37	41,69	57,10	72,51	$BRUSCO$
$s_{19,98,\ 22,49}$	−5,70	170,97	11	DEC	19,17	22,49	25,81	8,42	42,76	77,10	$BRUSCO$
$s_{22,49,\ 24,79}$	−2,05	88,87	12	DEC	23,90	24,79	25,68	32,93	38,04	43,16	$BRUSCO$
$s_{24,79,\ 26,64}$	1,70	−4,10	13	INC	25,30	26,64	27,98	38,88	41,19	43,50	$BRUSCO$
$s_{26,64,\ 28,83}$	7,85	−167,93	14	INC	27,90	28,83	29,75	42,82	58,35	73,89	$BRUSCO$
$s_{28,83,\ 31,41}$	−1,75	108,80	15	DEC	29,31	31,41	33,51	49,51	53,83	58,15	$BRUSCO$
$s_{31,41,\ 35,00}$	−3,77	172,25	16	DEC	36,00	35,00	37,00	39,40	40,30	41,20	$BRUSCO$

Tabla 5.1: Conjunto S de salida con $t_w = 3$ y $\epsilon_m = 5$.

s_{g_i,f_i}	$m_{f,g}$	$c_{f,g}$	**T**	$type_i$	t_i			v_i			$power_i$
$s_{0,00,\ 5,34}$	3,29	38,74	1	INC	5,27	5,34	5,41	55,17	56,31	57,45	$BRUSCO$
$s_{5,34,\ 6,86}$	−0,17	57,22	2	DEC	5,76	6,86	7,95	54,23	56,05	57,87	$MEDIO$
$s_{6,86,\ 11,92}$	−3,09	77,24	3	DEC	11,16	11,92	12,67	33,26	40,42	47,57	$BRUSCO$
$s_{11,92,\ 19,18}$	2,70	8,24	4	INC	18,28	19,18	20,08	58,05	60,03	62,00	$BRUSCO$
$s_{19,18,\ 19,25}$	0,70	46,60	5	INC	17,98	19,25	20,52	58,71	60,08	61,45	$BRUSCO$
$s_{19,25,\ 24,16}$	−5,04	157,12	6	DEC	22,62	24,16	25,69	18,09	35,37	52,65	$BRUSCO$
$s_{24,16,\ 29,42}$	3,98	−60,78	7	INC	27,83	29,42	31,01	45,69	56,31	66,94	$BRUSCO$
$s_{29,42,\ 35,00}$	−2,47	128,85	8	DEC	34,05	35,00	35,95	41,55	42,56	43,57	$BRUSCO$

Tabla 5.2: Conjunto S de salida con $t_w = 5$ y $\epsilon_m = 5$.

5.2. Número de Matrimonios Anuales en Escocia desde 1855 a 2011.

La fuente de los datos ha sido el *General Register Office for Scotland* a través de la dirección *http://www.gro-scotland.gov.uk/statistics/theme/vital-events/marriages-and-civil-partnerships/time-series.html*. Tiene información de los matrimonios anuales desde el año 1855 hasta el año 2011. En la experimentación se han utilizado todas las entradas, es decir, 157 registros. La Figura 5.4 muestra la serie de entrada. Se han realizado tres pruebas diferentes similares a las de la sección anterior, es decir, un tamaño de ventana t_w que ha tomado los valores 3, 5 y 7.

Los Conjuntos Ordenados de Segmentos S obtenidos tienen 44, 30 y 19 tendencias. Las Figuras 5.4, 5.5 y 5.6 muestran la comparativa de la serie de entrada con el resultado obtenido para cada una de las pruebas. El ECM obtenido para cada una de las pruebas es 1114,8121, 6409,4629 y 2410,1348. Finalmente, los Conjuntos Ordenados de Estructuras Difusas T que se obtienen son similares al caso anterior.

Esta prueba refrenda las conclusiones de la sección anterior para las dos representa-

s_{g_i,f_i}	$m_{f,g}$	$c_{f,g}$	**T**	$type_i$	t_i			v_i			$power_i$
$s_{0{,}00,\ 5{,}88}$	3,44	38,64	1	INC	5,52	5,88	6,24	58,71	58,88	59,06	$BRUSCO$
$s_{5{,}88,\ 12{,}23}$	−3,04	76,75	2	DEC	11,63	12,23	12,84	33,94	39,57	45,20	$BRUSCO$
$s_{12{,}23,\ 18{,}33}$	4,04	−9,85	3	INC	17,46	18,33	19,21	60,13	64,20	68,27	$BRUSCO$
$s_{18{,}33,\ 22{,}66}$	−4,92	154,48	4	DEC	22,11	22,66	23,20	32,90	42,89	52,89	$BRUSCO$
$s_{22{,}66,\ 35{,}00}$	0,64	28,39	5	INC	33,50	35,00	36,50	49,02	50,80	52,58	$BRUSCO$

Tabla 5.3: Conjunto S de salida con $t_w = 7$ y $\epsilon_m = 5$.

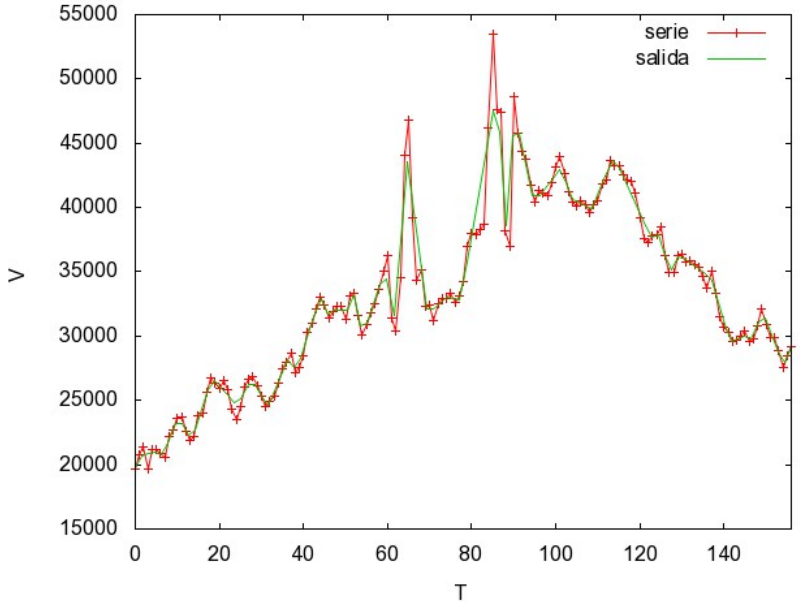

Figura 5.4: Test Número de Matrimonios Anuales en Escocia con $t_w = 3$.

ciones presentadas. Destacar que en la Figura 5.5 se observa nuevamente el fallo provocado por el tamaño de la ventana y se vuelve a confirmar que se debe trabajar para solucionar este problema. Este fallo provoca que el error cuadrático medio para $t_w = 5$ sea el mayor de todas las pruebas de este test, mientras que para $t_w = 3$ y $t_w = 7$ los resultados obtenidos son bastante satisfactorios.

5.3. Número de Divorcios Anuales en Escocia desde 1855 a 2011.

La fuente de los datos para esta prueba ha sido el *General Register Office for Scotland* a través de la dirección *http://www.gro-scotland.gov.uk/statistics/theme/vital-events/divorces-and-dissolutions/time-series.html*. Tiene información desde el año 1855 hasta el año 2011. En la experimentación se han utilizado todas las entradas. La Figura 5.7 muestra la serie de entrada. Al igual que en los dos casos anteriores, se han realizado tres pruebas diferentes utilizando un tamaño de ventana t_w que ha tomado los valores 3, 5 y 7.

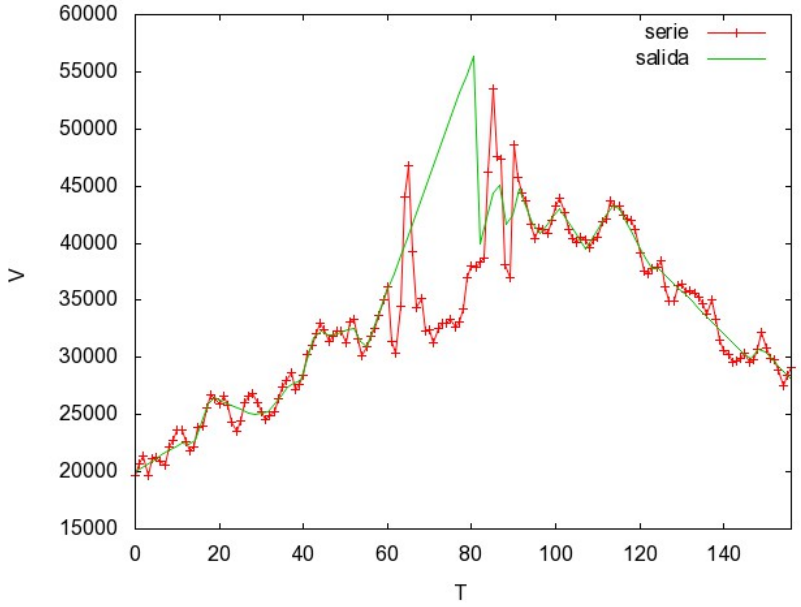

Figura 5.5: Test Número de Matrimonios Anuales en Escocia con $t_w = 5$.

Los Conjuntos Ordenados de Segmentos S obtenidos tienen 53, 28 y 19 tendencias. Las Figuras 5.7, 5.8 y 5.9 muestran la comparativa de la serie de entrada con el resultado obtenido para cada una de las pruebas. El ECM obtenido para cada una de las pruebas es $660{,}8805$, $442{,}1797$ y $500{,}6152$ respectivamente. Finalmente, los Conjuntos Ordenados de Estructuras Difusas T que se obtienen son similares al caso anterior.

Este test confirma las conclusiones de las dos test anteriores. En este caso no se observa el fallo provocado por el tamaño de la ventana, y S modela correctamente la serie de entrada. Los $ECMs$ son similares en las tres pruebas.

5.4. Análisis Criterios de Comparación.

Con los datos de la Sección 5.3 se ha evaluado la calidad de los criterios de comparación explicados en la Sección 4.3.2. Para realizar esta evaluación se han comparado los datos de los tres casos creados en función del tamaño de la ventana 3, 5 y 7, y los tres métodos de comparación expuestos: Binario, sin umbral y con umbral.

Para realizar la evaluación se ha utilizado consultas básicas como es la búsqueda de un pico (Ecuación 5.1).

$$c(-5{,}4, 215, 2), c(4{,}2, -48, 2) \tag{5.1}$$

En primer lugar se utilizará la búsqueda binaria. En la Tabla 5.4 se aprecian los resultados obtenidos en la comparación binaria entre los datos de la Sección 5.3 y la secuencia de búsqueda de la Ecuación 5.1. Se puede deducir de los resultados que este criterio no es nada bueno, puesto que es el que utilizan las bases de datos con datos numéricos y no es apropiado para las series temporales.

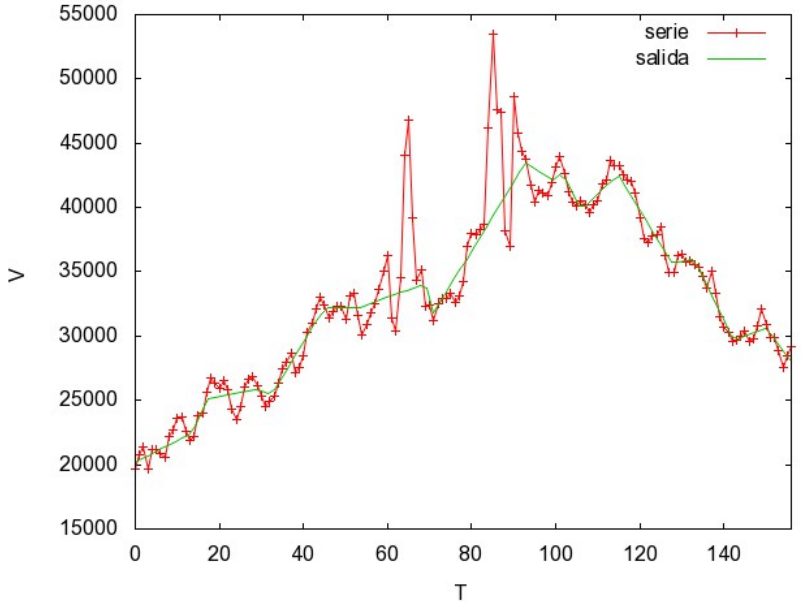

Figura 5.6: Test Número de Matrimonios Anuales en Escocia con $t_w = 7$.

Tamaño de ventana	Primeros resultados
3	r(0,d_{1-1},100),r(1,d_{1-2},100),r(2,d_{1-3},100)
5	r(0,d_{1-1},100),r(1,d_{1-2},100),r(2,d_{1-3},100)
7	r(0,d_{1-1},100),r(1,d_{1-2},100),r(2,d_{1-3},100)

Tabla 5.4: Resultado criterio de comparación binario.

Ahora se pasa a estudiar el caso de la comparación sin umbral. La Tabla 5.5 muestran los resultados de la consulta en la serie con este criterio. Se puede apreciar cómo según el tamaño de ventana el resultado es distinto. Esto se debe a que el preproceso que se explicó provoca que la forma de las señales no sea exactamente igual.

Tamaño de ventana	Primeros resultados
3	r(176,d_{1-1},0.05),r(667,d_{1-2},0.36),r(175,d_{1-3},1.38)
5	r(44,d_{1-1},1.94),r(28,d_{1-2},2.05),r(17,d_{1-3},2.08)
7	r(15,d_{1-1},0.91),r(14,d_{1-2},1.78),r(16,d_{1-3},1.79)

Tabla 5.5: Resultado criterio de comparación sin umbral.

A partir de estos primeros resultados se pueden extraer unas primeras conclusiones que son: (1) El criterio de comparación binario da resultados bastante malos. (2) La búsqueda sin umbral da un resultado aceptable. No obstante habrá que hacer más pruebas de este método realizando búsquedas complejas. (3) La búsqueda con umbral aunque teóricamente aporta ventajas respecto a la anterior hay que indagar en cómo se puede buscar ese valor de umbral que permita discriminar los resultados no aceptables. (4) Hay que investigar

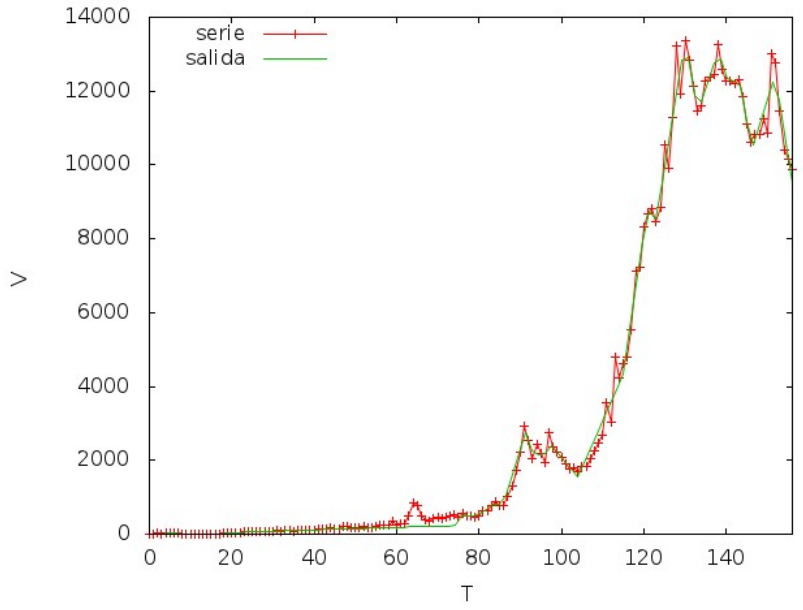

Figura 5.7: Test Número de Divorcios Anuales en Escocia con $t_w = 3$.

cómo integrar estos criterios de comparación en un gestor de base de datos real (postgresql, Mysql, ...).

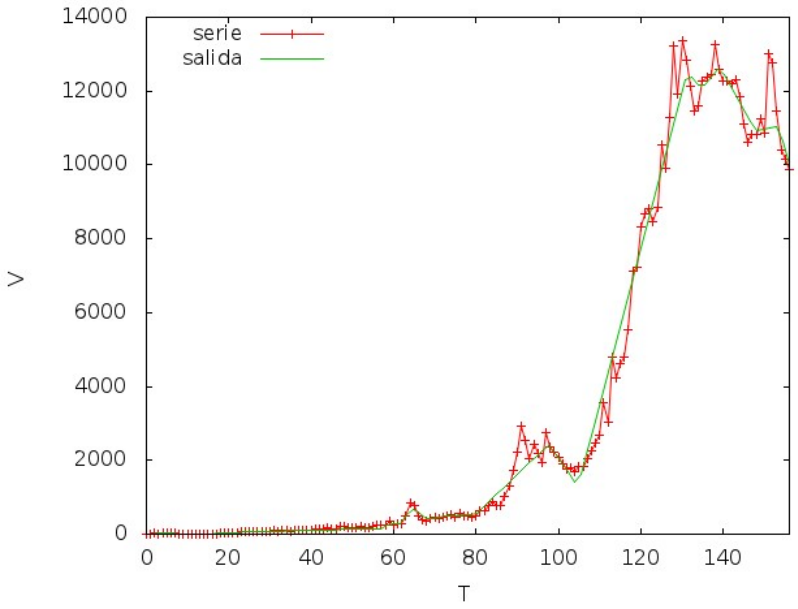

Figura 5.8: Test Número de Divorcios Anuales en Escocia con $t_w = 5$.

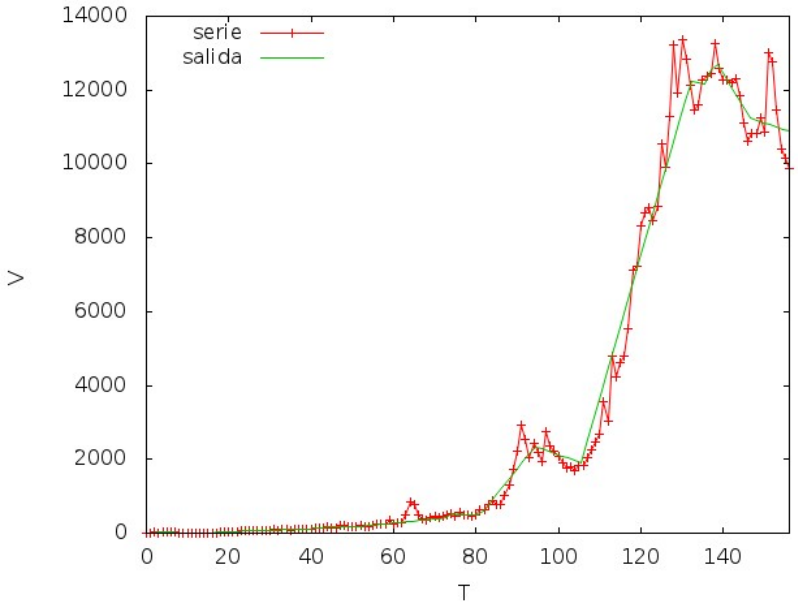

Figura 5.9: Test Número de Divorcios Anuales en Escocia con $t_w = 7$.

6

Conclusiones y Trabajos Futuros

En este trabajo se ha llevado a cabo un estudio de las series temporales. Más concretamente se ha estudiado la literatura más importante relativa a su representación, así como a la búsqueda o consulta en las mismas. Se han estudiado las representaciones por muestra, a trozos, con técnicas de compresión, mediante rectas, segmentos y las relativas a la lógica difusa. En concreto, se hace especial hincapié en las que usan segmentos y la lógica difusa ya que se consideran prometedoras para la realización de las consultas. También se ha realizado un repaso de las técnicas de comparación (Matemáticas, DTW, LCSS y EDT) y la literatura de interés en la que se mencionaban mejoras y posibles problemas detectados por algunos autores. También se han revisado algunas publicaciones que abordan la búsqueda en bases de datos de series temporales contemplando soluciones donde se generan índices, cluster y otras técnicas referentes a la optimización de consultas de base de datos.

Tras la situación del contexto en el estado del arte, se presentan dos propuestas de representación y una primera aproximación a las consultas sobre la primera representación. La primera representación está basada en segmentos obtenidos con regresión lineal, luego la serie se representa mediante un Conjunto Ordenado de Segmentos. No sólo se ha propuesto la representación sino que también se presenta un algoritmo para obtenerla de forma automática a partir de la serie. Este algoritmo utiliza dos parámetros. La segunda representación está basada en la lógica difusa. En concreto, se convierte la primera representación a Conjunto Ordenado de Estructuras Difusas. En este caso también se presenta el método para realizar la conversión. Respecto a las consultas se presenta una primera aproximación a las consultas sobre la representación como Conjunto Ordenado de Segmentos. Se introduce también una representación para las consultas sobre el Conjunto Ordenado de Segmentos. De los test realizados se puede concluir que ambas representaciones son robustas y coherentes, aunque se debe mejorar el algoritmo para la obtención de la serie como Conjunto de Segmentos para solucionar el problema visto en el Capítulo 5.

Como trabajo futuro se plantean diferentes líneas:

- Solucionar el problema visto en el Capítulo 5 para valores altos de t_w.

- Profundizar en la búsqueda con umbral para encontrar el valor de umbral que permita discriminar los resultados no aceptables.
- Diseño de métodos de consulta más complejos sobre el Conjunto Ordenado de Segmentos S.
- Diseño de métodos de consulta sobre el Conjunto Ordenado de Estructuras Difusas F.
- Comparación de los métodos propuestos que utilizan la lógica difusa con los que no para estudiar la efectividad o no de esta lógica.
- Diseño de un robusto conjunto de test que verifiquen la validez de los métodos propuestos y comparen con otros de los métodos estudiados en el Capítulo 3.

Bibliografía

[1] W.H. Woodall . Sullivan. A comparison of fuzzy forecasting and markov modeling. *Fuzzy Sets and Systems 64*, 3:279—-293, 1994.

[2] Gelb A. Applied optimal estimation. *MIT Press*, pages–, 1986.

[3] Haar A. Theorie der orthogonalen funktionen-systeme. *Mathematische Annalen*, 35:331–371, 1910.

[4] J. Abonyi, B. Feil, S. Nemeth, and P. Arva. Modified Gath–Geva clustering for fuzzy segmentation of multivariate time-series. *Fuzzy Sets and Systems*, 149:39–56, 2005.

[5] Faloutsos C. Swami A. Agrawal, R. Efficient similarity search in sequence databases. *Proceedings of the Fourth International Conference on Foundations of Data Organization and Algorithms*, pages 69–84, 1993.

[6] J. Wang Agrawal, J. Han and P. Yu. A framework for clustering evolving data streams. *Proc. 29th Very Large Data Bases Conf.*, pages–, 2003.

[7] D.A. Bao. Generalized model for financial time series representation and prediction. applied intelligence 29. 1:1–11, 2008.

[8] I.Z. Batyrshin and L.B. Sheremetov. Perception-based approach to time series data mining. *Applied Soft Computing*, 8:1211–1221, 2008.

[9] Clifford J. Berndt, D.J. Using dynamic time warping to find patterns in time series. *AAAI Working Notes of the Knowledge Discovery in Databases Workshop*, pages 359–370, 1994.

[10] Yazdani N. Ozsoyoglu Z.M. Bozkaya, T. Matching and indexing sequences of different lengths. *Proceedings of the Sixth ACM International Conference on Information and Knowledge Management*, pages 128–135, 1997.

[11] J. Pei X. Yan C. Giannella, J. Han and P.S. Yu. Mining frequent patterns in data streams at multiple time granularities. *Data Mining: Next Generation Challenges and Future Directions*, pages–, 2003.

[12] E. Egrioglu U. Yolcu V.R. Uslu C.H. Aladag, M.A. Basaran. Forecasting in high order fuzzy time series by using neural networks to define fuzzy relations. *Expert Systems with Applications 36*, 3:4228—-4231, 2009.

[13] S. Gunay U. Yolcu C.H. Aladag, E. Egrioglu. High order fuzzy time series forecasting model and its application to imkb. *Anadolu University Journal of Science and Technology 11*, 2:95—-101, 2010.

[14] H.J. Teoh C.H. Chiang C.H. Cheng, T.L. Chen. Fuzzy time series based on adaptive expectation model for taiex forecasting. *Expert Systems with Applications 34*, pages 1126—1132, 2008.

[15] J.W. Wang C.H. Cheng, G.W. Cheng. Multi-attribute fuzzy time series method based on fuzzy clustering. *Expert Systems with Applications 34*, 3:1235—-1242, 2008.

[16] Fu A. Yu C. Chan, K.P. Haar wavelets for efficient similarity search of time series: with and without time warping. *IEEE Transactions on Knowledge and Data Engineering 15*, 3:685–705, 2003.

[17] Kin-Pong Chan and Ada Wai-Chee Fu. Efficient time series matching by wavelets. *ICDE*, pages 126–133, 1999.

[18] Ozsu M.T. Oria V. Chen, L. Robust and fast similarity search for moving object trajectories. *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, pages 491–502, 2005.

[19] S.M. Chen. Forecasting enrollments based on fuzzy time series. *Fuzzy Sets and Systems 81*, 2:311–319, 1996.

[20] Hong Tzung Chen C. Fuzzy data mining for time-series data. *Applied Soft Computing*, pages 536–542, 2012.

[21] D.A. Chiang, L. R. Chow, and Y.F. Wang. Mining time series data by a fuzzy linguistic summary system. *Fuzzy Sets and Systems*, 112:419–432, 2000.

[22] M. Ranganathan Christos Faloutsos and Yannis Manolopoulos. Fast subsequence matching in time-series databases. *Proceedings 1994 ACM SIGMOD Conference, Mineapolis, MN*, pages 419–429, 1994.

[23] Keogh E. Hart D. Pazzani M. Chu, S. Iterative deepening dynamic time warping for time series. *Proceedings of the Second SIAM International Conference on Data Mining*, pages–, 2002.

[24] Lam S.K. Wong M.H. Chu, K.W. An efficient hash-based algorithm for sequence data searching. *The Computer Journal 41*, 6:402–415, 1998.

[25] F.L. Chung, T.C. Fu, Ng V., and Luk R.W.P. An Evolutionary Approach to Pattern-Based Time Series Segmentation. *Applied Soft Computing*, 8:1211–1221, 2008.

[26] Fu T.C. Luk R. Ng V. Chung, F.L. Flexible time series pattern matching based on perceptually important points. *International Joint Conference on Artificial Intelligence Workshop on Learning from Temporal and Spatial Data*, pages 1–7, 2001.

[27] Gunopulos D. Mannila H. Das, G. Finding similar time series. *Proceedings of the First European Symposium on Principles and Practice of Knowledge Discovery in Databases*, pages 88–100, 1997.

[28] P. D'Urso. Fuzzy Clustering for Data Time Arrays With Inlier and Outlier Time Trajectories. *IEEE Transactions on Fuzzy Systems*, 13(5):583–604, 2005.

[29] P. D'Urso and E.A. Maharaj. Autocorrelation-based fuzzy clustering of time series. *Fuzzy Sets and Systems*, 160:3565–3589, 2009.

[30] U. Yolcu V.R. Uslu N.A. Erilli E. Egrioglu, C.H. Aladag. Fuzzy time series forecasting method based on gustafson–kessel fuzzy clustering. *Expert Systems with Applications 38*, pages 10355—-10357, 2011.

[31] Manolopoulos Y Faloutsos C, Ranganathan M. Fast subsequence matching in time-series databases. *Proceedings of the ACM SIGMOD conference, Minneapolis, MN*, pages 419–429, 1994.

[32] Pratt K.B. Gandhi H.S. Fink, E. Indexing of time series by major minima and maxima. *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics,*, pages 2332–2335, 2003.

[33] Chung F.L. Luk R. Ng C.M. T.C. Chung F.L. Luk R. Ng C.M. Fu, T.C. Representing financial time series based on data point importance. *Engineering Applications of Artificial Intelligence 21*, 2:277–300, 2008.

[34] Keogh E. Lau L. Ratanamahatana C.A. Wong C.W. Fu, A. Scaling and time warping in time series querying. *The VLDB Journal 17*, 4:899–921, 2008.

[35] Che W.G. Zhao Q.J. Fu F.P., Chi K. High-order difference heuristic model of fuzzy time series based on particle swarm optimization and information entropy for stock markets. *International Conference on Computer Design and Applications*, 2010.

[36] Anguelov D. Indyk P. Motwani R. Gavrilov, M. Mining the stock market: which measure is best? *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,*, pages 487–496, 2000.

[37] Smyth P. Ge, X. Deformable markov model templates for time-series pattern matching. *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 81–90, 2000.

[38] Dina Q. Goldin and Paris C. Kanellakis. On similarity queries for time series data: Constraint specification and implementation. *Proceedings of the 1st International Conference on Principles and Practice of Constraint Programming*, pages–, 1995.

[39] Kanellakis P. Goldin, D. On similarity queries for time-series data:constraint specification and implementation. *Proceedings of the First International Conference on Principles and Practice of Constraint Programming*, pages 137–153, 1995.

[40] Kao T.W. Chen Y.H. Run R.S. Chen R.J. Lai J.L. Kuo I.H. Hsu L.Y., Horng S.J. Temperature prediction and taifex forecasting based on fuzzy relationships and mtpso techniques. *Expert Systems with Application 37*, pages 2756—-2770, 2010.

[41] Kao T.W. Run R.S. Lai J.L. Chen R.J. Kuo I.H. Khan M.K. Huang Y.L., Horng S.J. An improved forecasting model based on the weighted fuzzy relationship matrix combined with a pso adaptation for enrollments. *International Journal of Innovative Computing, Information and Control 7*, 7:4027—-4046, 2011.

[42] T.H.K. Yu Huarng K. The application of neural networks to forecast fuzzy time series. *Physica A 363*, pages 481—-491, 2006.

[43] T.H.K. Yu Huarng K. Ratio-based lengths of intervals to improve fuzzy time series forecasting. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics 36*, pages 328—-340, 2006.

[44] Yu T.H.K. Huarng K.H. A neural network-based fuzzy time series model to improve forecasting. *Expert Systems with Application 37*, pages 3366—3372, 2010.

[45] T.W. Kao T.L. Lin C.L. Lee Y. Pan I.H. Kuo, S.J. Horng. An improved method for forecasting enrollments based on fuzzy time series and particle swarm optimization. *Expert Systems with Applications 36*, pages 1494—-1502, 2009.

[46] Y.H. Chen R.S. Run T.W. Kao R.J. Chen J.L. Lai T.L. Lin I.H. Kuo, S.J. Horng. Forecasting taifex based on fuzzy time series and particle swarm optimization. *Expert Systems with Applications 37*, pages 1494—-1502, 2010.

[47] Astrom K. J. On the choice of sampling rates in parametric identification of time series. *Report 6807, Lund Institute of technology division of automatic control*, 1968.

[48] A. J. Owczarekb J. M. Leskia. A time-domain-constrained fuzzy clustering method and its application to signal analysis. *Fuzzy Sets and Systems*, 155:165–190, 2005.

[49] C.K. Song M.G. Chun J.I. Park, D.J. Lee. Taifex and kospi 200 forecasting based on two factors high order fuzzy time series and particle swarm optimization. *Expert Systems with Application 37*, 7:959—-967, 2010.

[50] Huarng K. Effective length of intervals to improve forecasting in fuzzy time series. *Fuzzy Sets and Systems 123*, pages 539—-548, 2001.

[51] Weierstrass K. Mathematische werke, volume ii. *Mayer and Muller,Berlin*, pages–, 1895.

[52] Chakrabarti K. Mehrotra S. Pazzani M. Keogh, E. Locally adaptive dimensionality reduction for indexing large time series databases. *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*, pages 151–163, 2001.

[53] Chakrabarti K. Pazzani M. Mehrotra S. Keogh, E. Dimensionality reduction for fast similarity search in large time series databases. *Journal of Knowledge and Information Systems 3*, 3:263–286, 2000.

[54] E. Keogh. A fast and robust method for pattern matching in time series databases. *Proceedings of the Ninth IEEE International Conference on Tools with Artificial Intelligence*, pages 578–584, 1997.

[55] E. Keogh. Exact indexing of dynamic time warping. *Proceedings of the 28th International Conference on Very Large Databases*, pages 406–417, 2002.

[56] Pazzani M. Keogh, E. Scaling up dynamic time warping for datamining applications. *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,*, pages 285–289, 2000.

[57] Pazzani M. Keogh, E. Derivative dynamic time warping. *Proceedings of the First SIAM International Conference on Data Mining.*, pages–, 2001.

[58] M. Khashei, S. Reza, and M. Bijari. A new hybrid artificial neural networks and fuzzy regression model for time series forecasting. *Fuzzy Sets and Systems*, 159:769–786, 2008.

[59] Park S.H. Chu W.W. Kim, S.W. An index-based approach for similarity search supporting time warping in large sequence databases. *Proceedings of the 17th IEEE International Conference on Data Engineering*, pages 607–614, 2001.

[60] Wong M.H.A. Lam, S.K. Fast projection algorithm for sequence data searching. *Data and Knowledge Engineering 28*, 3:321–339, 1998.

[61] Megalooikonomou V. Wang Q. Lakaemper R. Ratanamahatana C.A. Keogh E. Latecki, L.J. Partial elastic matching of time series. *Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 701–704, 2005.

[62] Kwon D. Lee S. Lee, S. Dimensionality reduction for indexing time series based on the minimum distance. *Journal of Information Science and Engineering 19*, pages 697–711, 2003.

[63] Govindaraju V. Lei, H. Regression time warping for similarity measure of sequence. *Proceedings of the Fourth International Conference on Computer and Information Technology*, pages 826–830, 2004.

[64] S.M. Chen L.W. Lee, L.H. Wang. Temperature prediction and taifex forecasting based on fuzzy logical relationships and genetic algorithms. *Expert Systems with Applications 33*, pages 539—-550, 2007.

[65] S.M. Chen L.W. Lee, L.H. Wang. Temperature prediction and taifex forecasting based on high-order fuzzy logical relationships and genetic simulated annealing techniques. *Expert Systems with Applications 34*, pages 328—336, 2008.

[66] BP.F. Marteau. Time Warp Edit Distance with Stiffness Adjustment for Time Series Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):306–318, 2009.

[67] P. Melin, M. Mancilla, A.and Lopez, and O. Mendoza. A hybrid modular neural network architecture with fuzzy Sugeno integration for time series forecasting. *Applied Soft Computing*, 7:1217–1226, 2007.

[68] F. Montesino, A. Lendassea, and A. Barriga. Autoregressive time series prediction by means of fuzzy inference systems using nonparametric residual variance estimation. *Fuzzy Sets and Systems*, 161:471–497, 2010.

[69] F. Morchen. Time series feature extraction for data mining using dwt and dft. *University of Marburg, Department of Mathematics and Computer Science, Technical Report no. 33.*, pages–, 2003.

[70] Patel J.M. Morse, M.D. An efficient and accurate method for evaluating time series similarity. *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, pages 569—580, 2007.

[71] R. Schneider N. Beckmann, H.-P. Kriegel and B. Seeger. The r*-tree: An efficient and robust access method for points and rectangles. *Proceedings of ACM SIGMOD Int?l. Conf. on Management of Data*, pages 322–331, 1990.

[72] Fink-E. Pratt, B. Search for patterns in compressed time series. *International Journal of Image and Graphics 2*, 1:89–106, 2002.

[73] B.S. Chissom Q. Song. Forecasting enrollments with fuzzy time series - part i. *Fuzzy Sets and Systems 54*, pages 1–9, 1993.

[74] B.S. Chissom Q. Song. Fuzzy time series and its models. *Fuzzy Sets and Systems 54*, pages 269–277, 1993.

[75] B.S. Chissom Q. Song. Forecasting enrollments with fuzzy time series - part ii. *Fuzzy Sets and Systems 54*, pages 1–8, 1994.

[76] B.S.Leland Q. Song. Adaptative learning defuzzification techniques and applications. *Fuzzy Sets and Systems 54*, pages 321–329, 1996.

[77] H. S. Sawhney R. Agrawal, K. I. Lin and K. Shim. Fast similarity search in the presence of noise, scaling, and translation in times-series databases. *Proceedings of 21rst International Conference on Very Large Data Bases*, pages 490–500, 1995.

[78] Davood Rafiei and Alberto O. Mendelzon. Similarity-based queries for time series data. pages 13–25, 1997.

[79] Davood Rafiei and Alberto O. Mendelzon. Efficient retrieval of similar time sequences using dft. *FODO*, pages–, 1998.

[80] C.A. Ratanamahatana and E.R Keogh. Making time-series classification more accurate using learned constraints. *Proceedings of the Fourth SIAM International Conference on Data Mining*, pages 11–22, 2004.

[81] Keogh E. Ratanamahatana, C.A. Three myths about dynamic time warping data mining. *Proceedings of the Fifth SIAM International Conference on Data Mining.*, pages–.

[82] Keogh E. Bagnall A.J. Lonardi S.A Ratanamahatana, C.A. Novel bit level time series representation with implications for similarity search and clustering. *Proceedings of the Ninth Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 771–777, 2005.

[83] H.F. Wang F R.C. Tsaur, J.C. Yang. Fuzzy relation analysis in fuzzy time series model. *Computers and Mathematics with Application 49*, 2:539—-548, 2005.

[84] Niennattrakul V. Ratanamahatana C.A. Ruengronghirunya, P. Speeding up similarity search on a large time series data set under time warping distance. *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, pages 981–988, 2009.

[85] I.B. Turksen S. Davari, M.H.F. Zarandi. An improved fuzzy time series forecasting model based on particle swarm intervalization. *The 28th North American Fuzzy Information Processing Society Annual Conferences*, pages 647—-651, 2009.

[86] Yoshikawa M. Faloutsos C Sakurai, Y. Ftw: fast similarity search under the time warping distance. *Proceedings of the 24th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 326–337, 2005.

[87] Chan P. Salvador, S. Fastdtw: toward accurate dynamic time warping in linear time and space. *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining Workshop on Mining Temporal and Sequential Data*, pages 70–80, 2004.

[88] G.A.F. Seber and A.J. Lee. *Linear Regression Analysis*. Wiley-interscience, 2003.

[89] Mamoulis N. Cheung D.W. Shou, Y. Fast and exact warping of time series using adaptive segmental approximations. *Machine Learning 58*, pages 231–267, 2005.

[90] S.R. Singh. A simple method of forecasting based on fuzzy time series. *Applied Mathematics and Computation 186*, 3:330—-339, 1965.

[91] H. Song, C. Miao, W. Roel, Z. Shen, and F. Catthoor. Implementation of Fuzzy Cognitive Maps Based on Fuzzy Neural Network and Application in Prediction of Time Series. *IEEE Transactions ont Fuzzy Systems*, 18(2):233–250, 2010.

[92] S.Y. Lin S.T. Li, Y.C. Cheng. Fcm-based deterministic forecasting model for fuzzy time series. *Computers and Mathematics with Applications 56*, pages 3052—-3063, 2008.

[93] W. Stach, L.A. Kurgan, and W. Pedrycz. Numerical and Linguistic Prediction of Time Series With the Use of Fuzzy Cognitive Maps. *IEEE Transactions ont Fuzzy Systems*, 116:61–72, 2008.

[94] E. Egrioglu V.R. Uslu U. Yolcu, C.H. Aladag. Time series forecasting with a novel fuzzy time series approach: an example for istanbul stock market. *Journal of Statistical Computation and Simulation*, pages 647—-651.

[95] V.R. Uslu M.A. Basaran C.H. Aladag U. Yolcu, E. Egrioglu. A new approach for determining the length of intervals for fuzzy time series. *Applied Soft Computing 9*, pages 647—-651, 2009.

[96] Zwir I.S. uspini, E.H. Automated qualitative description of measurements. *Proceedings of the 16th IEEE Instrumentation and Measurement Technology Conference.*, pages–, 1999.

[97] O. Valenzuela, I. Rojas, F. Rojas, H. Pomares, L.J. Herrera, A. Guillen, L. Marqueza, and M. Pasadas. Hybridization of intelligent techniques and ARIMA models for time series prediction. *Fuzzy Sets and Systems*, 159:821–845, 2008.

[98] Gunopulos D. Kollios G. Vlachos, M. Discovering similar multidimensional trajectories. *Proceedings of the 18th IEEE International Conference on Data Engineering*, 2:673—-684, 2002.

[99] K.C.C. Chan W.H. Au. Mining fuzzy rules for time series classification. *The 2004 IEEE International Conference on Fuzzy Systems*, 1:239–244, 2004.

[100] Faloutsos C. Sycara K. Payne T.R. Wu, L. Falcon: feedback adaptive loop for content-based retrieval. *Proceedings of the 26th International Conference on Very Large Databases*, pages 297–306, 2000.

[101] G. Dong Y. Chen. Multi-dimensional regression analysis of time-series data streams. *Proc. 2002 Int?l Conf. Very Large Data Bases (VLDB 02), 2002.*, pages–, 2002.

[102] Kyu-Young Whang Yang-Sae Moon and Woong-Kee Loh. Efficient time-series subsequence matching using duality in constructing windows. *nformation Systems, 26*, 4:279–293, 2001.

[103] C.H. Lee Y.K. Bang. Fuzzy time series prediction using hierarchical clustering algorithms. *Expert Systems with Applications 38*, pages 4312—-4325, 2011.

[104] H.K. Yu. Weighted fuzzy time series models for taiex forecasting. *Physica A 349*, 3:609—624, 2005.

[105] Zhang S. Zhao, Y. Generalized dimension-reduction framework for recent-biased time series analysis. *IEEE Transactions on Knowledge and Data Engineering 18*, 2:–, 2006.

[106] Wong M.H.A. Zhou, M. Segment-wise time warping method for time scaling searching. *Information Sciences 173*, pages 227–254, 2005.

[107] Shasha D. Zhu, Y. Warping indexes with envelope transforms for query by humming. *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, pages 181–192, 2003.

A

Currículum Vitae

A.1. Experiencia Profesional

Julio 1994 - Enero 1996 Beca de formación. Centro de calculo. Universidad Castilla La-Mancha. Albacete.

Marzo 1996 - Enero 1997 Admistrador de Base de datos. Centro de calculo. Universidad Castilla La-Mancha. Albacete.

Febrero 1998 - Noviembre 1999 Administrador de Base de Datos.Hospital General de Albacete. Albacete.

Noviembre 1999 - Agosto 2000 Programación proyecto europeo I.S.L.A. Instituto de Desarrollo Regional. Albacete.

Agosto 2000 - Diciembre 2001 Programación proyecto CICYT. Instituto de Desarrollo Regional. Albacete.

Diciembre 2001 - Mayo 2002 Programación proyecto CEDERCAM. Instituto de Desarrollo Regional. Albacete.

Diciembre 2002 - Marzo 2003 Técnico analista programador. S. M. Consultores S.L.. Madrid.

Abril 2003 - Mayo 2004 Programación proyecto CEDERCAM. Instituto de Desarrollo Regional. Albacete.

Mayo 2004 - Septiembre 2008 Técnico analista programador. S. M. Consultores S.L.. Madrid.

Septiembre 2005 - Agosto 2011 Tutor de Sistemas de Información Geográfica. Universidad Nacional Educación a Distancia. Albacete.

Septiembre 2008 - Actualidad Profesor de Secundaria. JCCM.

A.2. Publicaciones

"INTRODUCCIÓN A BASES DE DATOS GEORREFERENCIADAS", capítulo dentro del libro colectivo Especialista en Información Geográfica y Teledetección. Sistemas de Información Geográfica, UCLM, Albacete, 2001.

"CATASTRO DIGITAL", capítulo dentro del libro colectivo Especialista en Información Geográfica y Teledetección. Sistemas de Información Geográfica, UCLM, Albacete, 2001.

"PERSONALIZACIÓN DE APLICACIONES SIG", capítulo dentro del libro colectivo Especialista en Información Geográfica y Teledetección. Sistemas de Información Geográfica, UCLM, Albacete, 2001.

"ISLA".Land and water management in mediterranean islands using earth observation data.,UCLM,Albacete,2001.